Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2016-014 November 8, 2016

Report on the 2015 NSF Workshop on Unified Annotation Tooling

Finlayson, Mark Alan

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Report on the 2015 NSF Workshop on Unified Annotation Tooling

Mark A. Finlayson

Florida International University

&

Massachusetts Institute of Technology

November 8, 2016

Abstract

On March 30 & 31, 2015, an international group of twenty-three researchers with expertise in linguistic annotation convened in Sunny Isles Beach, Florida to discuss problems with and potential solutions for the state of linguistic annotation tooling. The participants comprised 14 researchers from the U.S. and 9 from outside the U.S., with 7 countries and 4 continents represented, and hailed from fields and specialties including computational linguistics, artificial intelligence, speech processing, multi-modal data processing, clinical & medical natural language processing, linguistics, documentary linguistics, sign-language linguistics, corpus linguistics, and the digital humanities. The motivating problem of the workshop was the balkanization of annotation tooling, namely, that even though linguistic annotation requires sophisticated tool support to efficiently generate high-quality data, the landscape of tools for the field is fractured, incompatible, inconsistent, and lacks key capabilities. The overall goal of the workshop was to chart the way forward, centering on five key questions: (1) What are the problems with current tool landscape? (2) What are the possible benefits of solving some or all of these problems? (3) What capabilities are most needed? (4) How should we go about implementing these capabilities? And, (5) How should we ensure longevity and sustainability of the solution? I surveyed the participants before their arrival, which provided significant raw material for ideas, and the workshop discussion itself resulted in identification of ten specific classes of problems, five sets of most-needed capabilities. Importantly, we identified annotation project managers in computational linguistics as the key recipients and users of any solution, thereby succinctly addressing questions about the scope and audience of potential solutions. We discussed management and sustainability of potential solutions at length. The participants agreed on sixteen recommendations for future work. This technical report contains a detailed discussion of all these topics, a point-by-point review of the discussion in the workshop as it unfolded, detailed information on the participants and their expertise, and the summarized data from the surveys.

Table of Contents

1. Problem

Computational linguistics, and especially its sub-area of statistical natural language processing (NLP), is a field bursting with new, important, and influential scientific and technical work. Importantly, much of this work has been enabled by linguistically annotated corpora: collections of linguistic artifacts (such as text, speech, or video) that have been marked up for some language phenomenon. Indeed, in just the past five years, there have been a number of prominent technological advances and innovations that have had broad impact on society that would not have been possible without numerous linguistically annotated corpora. As just three examples, witness Apple's Siri or Microsoft's Cortana for speech understanding, Google Translate for automatic language translation, and IBM's Watson for playing the Jeopardy game show. Large annotated corpora are a key resource that enables these advances, and they are fundamental to progress in the field.

But despite the widely-recognized importance of annotated corpora, the field has a major problem: it lacks coherent and functional software tool support. Collecting linguistically annotated data is a complex, labor-intensive, and difficult endeavor, and sophisticated software tools are required at every stage of the process. While there are hundreds of tools available right now, they suffer from three related problems:

Problem 1: Functionality. Tools do not provide the needed functionality, or do not provide it in an easily usable form. There are a number of tasks common across almost every linguistic annotation project, and many of these tasks have no or inadequate software support. Individual annotation projects usually involve more specialized tasks which often have even less software support. Even if tools do provide the needed functionality, the functionality is often hard to use or not documented properly.

Problem 2: Interoperability. Tools do not work together well. No tool can do everything, and so any linguistic annotation project must assemble tools together into pipelines, tool chains, and workflows. Despite this inescapable fact, tools often do not share input or output formats, do not use the same linguistic conceptual schemes, do not use the same terminology to describe their operation, and do not support the full range of annotation project types.

Problem 3: Reusability. Tools and resources cannot be easily applied to new problems. Tools are not extensible in a way that allows new functionality to be added, and the conceptual and theoretical schemes underlying the linguistic annotation itself is usually not specified precisely enough to allow other researchers to build upon them. Use and reuse of annotated tooling are enabled by clear specifications and functional software tooling; missing, poorly designed, or poorly specified tools block reuse.

In general, I have called these three problems together the **balkanization of annotation tooling**. Faced with this problem, many language researchers create their own tools from scratch, at significant cost. These tools are usually hastily designed, not released for general use, not maintained, and often redundant with capabilities implemented by other tools. Researchers end up building tools that do the same things time and again. This adds up to great waste of time, money, and effort, and blocks progress in the field. The goal of the workshop described here is to plan an approach to addressing these problems.

1.1. An Analogy

To drive home the idea that problems with tool support have a detrimental effect on work in linguistic annotation, consider an analogy to a domain with which most of us are familiar: writing an electronic document. If one wants to write an electronic document right now, what does one need to do? You open up a common word processing application (e.g., Microsoft Word, Google Docs, OpenOffice) and start typing. It's simple. If you don't like your word processor, you can download and install a different one in a minute from a small handful of options, and they are all about as easy to use and support much of the same basic functionality.

Now, imagine that writing an electronic document were like linguistic annotation, and a particular document were like the raw language data. What would it be like if the state of word processing were the same as linguistic annotation today?

First, your machine would not have any word processors installed, so you would first try to find a ready-made word processor to write the document. You would have a hundred or more different options, all poorly documented and in various states of disrepair and availability. You would read a bunch of academic articles, trying to find the one word processor that supported the type of document you wanted to write. Some word processing applications would sound promising, only for you to find that they are no longer available, or do not run on modern operating systems. Other

word processors you would download and try to run, but there would be an inexplicable error not covered in the documentation, and so you would spend some hours trying to get it to run by installing other support programs or adjusting configuration files.

Finally, in most cases, you would give up, and sit down to write your own word processor. You would spend quite a bit of time figuring out exactly which features of the document you needed to support: Italic font? Headers? Bulleted Lists? Perhaps your tool is tailored to the specific kind of document you are writing, say, a job application letter, or a romance novel, or a technical manual. Days, weeks or months of design, implementation, and bug fixes later, you have a tool which allows you to create your document.

Once you have created the document you tuck your newly-created tool away somewhere on your hard drive and never think about it again. When the next poor researcher comes along who wants to create a similar sort of document, they have to repeat the whole process, and often without the benefit of the insights you gained when designing and building your tool, because, for the most part, people don't publish papers describing the *ad hoc* one-off tools they build.

Think about this scenario for a minute. What effect would it have on your writing habits? Or on your productivity for tasks involving writing documents? Would writing electronic documents be something in which you would casually engage? Would you be able to easily collaborate on documents with others? Indeed, would writing documents be a task that could be readily repeated by others?

1.2. Detrimental Effects

The lack of functional, interoperable, and reusable tool support for linguistic annotation results in a constellation of major detrimental effects. It **reduces or blocks** the ability of researchers to build upon and replicate other's work. If a language resource requires sophisticated software support to build and use, then difficulties with or absence of that software support will naturally impede use or replication of that resource. The harder a tool is to use, and the more researchers must fight with poorly designed tools or create their own tools from scratch, the less likely it is they will pursue any particular research project. This leads to **lost opportunities**, as researchers forego projects that present too many difficulties in tool design. This means that many scientifically valid and worthwhile projects are not pursued, or their pursuit is **delayed or diminished**. Researchers spend significant amounts of time, money, and effort understanding, using, and implementing annotation tools. These are resources that could be better spent on advancing the field directly, rather that revisiting already trodden ground. This duplication of effort represents a significant **waste of resources**.

1.3. Definition of Terms

Throughout this report I use a number of terms that I will define here.

Annotation spans a spectrum along several dimensions and many fields of research use methods that could be described accurately using this term. In this report I focus on annotation that supports research and development on the computational understanding and use of language. In this context, one can define annotation to be "any descriptive or analytic notations applied to raw language data" [1].

An **annotation scheme** is a specification that defines the syntax and semantics of the linguistic phenomenon of interest by specifying appropriate labels and features for the annotation objects and relations among them. A new annotation project may use an existing scheme, either "as is" or with modification, or it may require development of a new scheme for phenomena that have not been previously annotated. Example annotation schemes include the Treebank II scheme [2], PropBank [3], and TimeML [4]. An annotation scheme can be split into the logical description of notations and how they are indexed to the raw language data and a human-readable **annotation guide** that describes how the annotations should be applied in practice such as [5].

An **annotation tool** supports annotation efforts and is, in its most narrow interpretation, a piece of computer software that facilitates identification of segments of the linguistic data along with the means to associate these segments with the relevant labels. Example annotation tools include GATE [6], Ellogon [7], or Brat [8]. Annotation projects also require all sorts of other software to support creation of the annotated data; to the degree this software is specialized to language annotation I will refer to them as annotation tools as well, where the context does not beget confusion.

An **annotated corpus**, then, is a collection of linguistic artifacts (text, speech, multi-modal video, etc.), that has had linguistic annotations associated with it conforming to one more annotation schemes. Example annotated corpora include the Penn Treebank [2], the OANC[1], MASC[2], and OntoNotes [9].

1.4. Summary of Results

With these problems in mind, I organized a workshop on unified annotation tooling (henceforth, the **UAT workshop**) to discuss and plan for future work.

I start by giving detail on the scientific context (§2.1), and the funding context (§2.2). I discuss the original goals of the workshop when planning began (§3.1), which transformed into revised goals and questions after many discussions with potential participants (§3.2). The driving questions of the workshop were to identify the problems in detail and to chart way forward, centered on answering 5 key questions: (1) What are the **problems** with current tool landscape? (2) What are the possible **benefits** of solving some or all of these problems? (3) What **capabilities** are most needed? (4) How should we go about **implementing** these capabilities? And, (5) How should we ensure **longevity and sustainability** of the solution?

Because the discussion and eventual recommendations of the workshop—which problems are considered important, what approaches are considered feasible and what theoretical considerations are valid—are heavily influenced by the participants, I gave careful consideration to **who to invite** so as to provide a representative and diverse set of views (§4).

I surveyed these participants in a pre-workshop survey and this set the stage for discussion, generating a significant amount of raw material to stimulate the participants (§5.1). I give a detailed, point-by-point explanation of the discussion (§5.2), with the goal of reflecting, as accurately as possible, how it unfolded during the workshop. To facilitate the accuracy of this explanation, I recorded the workshop discussions and made transcripts after the fact. I extracted the summary from transcripts, guided by the notes that I took at that time, which hopefully will allow readers of this report to see the depth of the discussion on various issues, and to potentially make up their own minds as to what conclusions should be drawn from the discussion.

The participants identified **ten classes of problems** for the annotation tool landscape (§5.2.1). We further prioritized these problems and fleshed them out in detail (§5.2.2). Because annotation is important to many fields of inquiry, it was important to address which target audience or discipline would take precedence (if any), so as to properly scope the recommendations. The **target audience** identified, after much discussion, was linguistic annotation project managers (§5.2.3).

To flesh out what capabilities were needed to overcome the various specific issues arising from specific annotation projects, the participants organized into breakout groups to brainstorm possible use cases (§5.2.4). This discussion produced several interesting ideas that lent depth and richness to later points and helped the group prioritize capabilities and audiences.

On the beginning of the second day, Erhard and Marie Hinrichs from CLARIN in Europe gave some context of the European approaches (§5.2.5). The discussion then turned to capabilities and we identified **five general classes of capabilities** that were motivated by the problems (§5.2.6). In general, with regard to implementation, it was agreed that **no one tool solves all problems** and that we must focus on reusing existing tools, adapting them appropriately, filling in the gaps, and facilitating interoperability.

One of the strongest recommendations of the workshop, which merited its own extensive discussion, was to propose a separate effort and **technical workshop** to promote interoperability among the various linguistic pipelines (§5.2.8).

The discussion also produced many good ideas with regard to **sustainability** and **partnerships** to ensure longevity of any effort to solve these problems (§5.2.7). However, participants acknowledged that this was a problem which did not have a clear solution, and depended on a large degree to whether researchers as a whole adopt particular solutions and lend their weight.

[1] http://www.anc.org/OANC
[2] http://www.anc.org/MASC

In the end we generated **sixteen recommendations** for moving the field forward in a positive direction (§5.2.9). These recommendations covered planning, describing & cataloging, implementation, and other activities, and are the major deliverable of this report.

2. Context

2.1. Scientific Context

2.1.1. Importance of Corpus Annotation

In just the past decade computational linguistics has produced a slate of new technologies that have changed what people have come to expect from their interactions with computers and language data. By way of example, think of the following three technologies: Siri, Apple's voice-recognition interface which was a merging of statistical voice-recognition and question-answering technologies; Watson, IBM's Jeopardy-playing computer, which handily beat two Jeopardy world-champions, a feat that was considered impossible not five years ago; and Google translate, which now provides a service, for free, that was once available only by fee via by highly-trained experts. These widely-hailed advances are driven by the near universal hunger for the analysis and use of more and more language data. The internet, and especially the web, is making electronically-accessible language data ever more voluminous: think of Google, Wikipedia, online newspapers, and the fact that nearly all written work produced in the present day is made available in electronic form. The now near-universal availability of powerful computers at people's fingertips makes this data more available and easy to analyze: nearly everyone has multiple computing devices, including a smartphone in their pocket and a laptop at home; businesses are equipped with personal workstations for nearly every employee and leverage huge datacenters full of processing power and storage space. All of these realities have led to the expectation, and the demand, that language data be used to perform broadly useful tasks such as finding information in huge databases, analyzing hidden statistical relationships, and enabling more natural human-computer interactions.

Underlying all of these advances in using language are major advances in statistical natural language processing (NLP); and underlying all the advances in NLP are **linguistically annotated corpora**. Annotated corpora are a key resource for computation linguistics, a *sine qua non* for nearly every language technology developed in the last thirty years. Starting with early resources such as the Brown corpus [10] and the Penn Treebank [2] and coupled with advances in computational power to perform sophisticated machine learning, annotated corpora have served as the foundation for investigations of scientific hypotheses regarding language, for discovering important regularities in language use, and for training statistical models for automatic language understanding. Without annotated corpora, we would be unable to talk to our computers and our phones, we would be unable to search huge databases for needles in the language haystack, and we would be unable to browse webpages, articles, and books written in different languages with the ease of clicking a button to automatically translate them.

2.1.2. A Missing Capability

Despite the critical importance of annotated corpora, we lack a critical capability: we have no consistent, interoperable, generalizable, easy-to-use tool support for performing linguistic annotation.

There are plenty of tools, of course. A brief survey of papers in recent computational linguistics conferences will reveal that a large fraction of the papers use annotated corpora to either train systems or evaluate results. Of those papers that use annotated corpora, a number will have created a corpus especially for that work, adding to the already rich set of corpora available. What tools do they use to create the corpora? Often they do not use an off-the-shelf tool. Rather, many—if not most—write their own tools from scratch. Because researchers in general do not have expertise in annotation tool design, the tools tend to be hastily designed and *ad hoc*. They are rarely released for public use, even when the corpus created with the tool is released. Even if the tools are released, they often do not operate with existing tools, do not conform to standards or best practices, lack documentation, and are not maintained in the future. All of this adds up to a significant waste of resources spent on building tools that accomplish a specific, one-off task and do not serve the field in the long term.

The lack of rational tool support for annotation leads to another problem: difficulty in visualizing, analyzing, and building upon existing annotated corpora. If a corpus is released without tool or API support, then accessing that data requires a researcher write their own software for that task. If a researcher wants to replicate or augment an

existing corpus, lack of access to the original tool (or the tool's limitations) makes this a daunting endeavor. If all a researcher wants to do is a simple annotation task, perhaps with a straightforward modification not supported by the available tools, having to build a tool just to create the corpus can have a chilling effect on the research.

2.1.3. A Selection of Related Efforts

That there are significant problems with the linguistic annotation tool landscape is not a new observation. A number of projects have attempted to tackle this or related problems, and while each has had success to a certain degree, the landscape still remains full of problems. I list below a (non-representative) selection of projects that have tried to tackle problems in linguistic annotation tooling. Importantly, I identified this initial selection early in planning for the workshop, and this provided a set of researchers who were potential attendees to the workshop. The projects selected show a range of different approaches:

- Comprehensive tools that attempt to solve a class or multiple classes of linguistic annotation problems, or were envisioned as "one size fits all" tools (e.g., GATE, Ellogon, the Story Workbench)
- Streamlined, open-source tools that seek to make annotation more like using a Word Processor or website, and put the annotator first (the Story Workbench, Brat)
- Architectures that seek to provide a way of connecting together existing tools (e.g., Gate, UIMA, the LAPPS Grid)
- Research federations that seek to streamline sharing, chaining, and reuse of resources, including tools (e.g., the LAPPS Grid, CLARIN)

GATE: General Architecture for Text Engineering [6] Sometimes referred to as the "MATLAB of NLP", GATE is a closely integrated set of tools for solving several related sets of problems in linguistic annotation. Its core strength is in setting up complex chains of automatic analyzers, via the GATE Developer Integrated Development Environment (IDE). It has additional tools for managing other aspects of the annotation process, such as manual annotation [11], crowdsourcing [12], or embedding GATE into other applications [6].

Ellogon [7] Developed in Greece, Ellogon is much like GATE in that it provides a general architecture to support natural language processing applications. It focuses on the text engineering and processing associated with annotated texts, and not on the annotator.

Story Workbench [13], [14] The Story Workbench is a tool for doing semi-automatic linguistic annotation on short texts. In contrast to GATE or Ellogon, the Story Workbench puts the annotator first, making ease-of-use for the annotator a key feature. The Story Workbench is installed locally, documents are managed and modified locally, and a source-control repository is used to share work with the team. It is focused on a "double-annotated plus adjudication" workflow for annotation. Like GATE and Ellogon it is built on a plugin model, in an effort to provide ease of extensibility.

Brat Rapid Annotation Tool [8] Brat also focuses on the annotator, providing a simple, easy-to-use web-based interface. In contrast to Story Workbench, Brat is installed on a webserver, and the annotator interacts exclusively with the tool through a web browser. Processing in Brat happens in the web-browser, and documents are stored remotely. Because of its remote nature, it requires no installation for the individual annotator, has a straightforward and easy-to-use user interface, and is well-documented. On the other hand, Brat does not provide a sophisticated backend programming model, is not easily extended (it is not a plugin-based architecture), and does not support local-storage or modification of documents, or any workflow that might require such a capability. Its graphical interface is intuitive and easy for sparse, local annotations.

UIMA: Unstructured Information Management Architecture[3] Originally developed by IBM, and now spun off into an Apache project, UIMA is a general framework for constructing chains of automatic analyzers. It is sophisticated, but does not directly address the problem of annotation by human annotators. UIMA implements several interesting features, in particular, a robust type system for describing the formal syntax of linguistic annotations and a general API for indexing annotations to documents of almost any nature. Many linguistic annotation projects require annotations to be applied not only to text, but to non-textual objects that themselves may contain language, such as images, scanned document pages, movies, and audio.

[3] https://uima.apache.org/

LAPPS: The Language Application Grid [15] The LAPPS project, funded by NSF, sought to create "an interoperable software infrastructure to support natural language processing (NLP) research and development. LAPPS built on the prior SILT project, the stated goal of which was "to turn existing, fragmented technology and resources developed to support language processing technology into accessible, stable, and interoperable resources that can be readily reused across several fields." The focus of this project was on standards for interoperability, and LAPPS established a large international collaborative effort involving key players from the U.S., Europe, Asia, and Australia. The project developed an open, web-based infrastructure through which distributed language resources can be easily accessed, in whole or in part, and within which tailored language services can be efficiently composed, disseminated and consumed by researchers, developers, and students.

CLARIN: Common Language Resources and Technology Infrastructure[4] The goal of this EU-funded CLARIN project is to provide easy and sustainable access for scholars in the humanities and social sciences to digital language data (in written, spoken, video or multimodal form) and advanced tools to discover, explore, exploit, annotate, analyze or combine them, wherever they are located. CLARIN is building a networked federation of language data repositories, service centers and centers of expertise, with single sign-on access for all members of the academic community in all participating countries. Tools and data from different centers are interoperable, so that data collections can be combined and tools from different sources can be chained to perform complex operations to support researchers in their work.

In addition to these efforts identified before the workshop, a number of other efforts were identified during the discussion, as noted in §5.

2.2. Funding Context

While the scientific context clearly indicates a problem and a need, it remained to be determined how the need should be addressed. As in much of science today, progress depends on securing funding to pay for the work. I noted that there exists a funding program—CISE Research Infrastructure, or CRI—within the U.S. National Science Foundation (NSF), which seemed to be a good fit to the problem at hand.

2.2.1. NSF CISE Research Infrastructure Program

The CISE Research Infrastructure (CRI) program[5] is a directorate-level program situated in the Computer & Information Science & Engineering Directorate (CISE). The goal of the program is to drive discovery and learning in the core CISE disciplines by supporting the creation and enhancement of world-class computing research infrastructure. The intention of these infrastructures is to enable CISE researchers to advance the frontiers of CISE research.

At the time when this work began, the CRI program supported two broad types of award: Institutional Infrastructure (II) and Community Infrastructure (CI). Institutional Infrastructure awards were intended for the creation or enhancement of research infrastructure at specific institutions. Community Infrastructure (CI) awards were intended to support the creation of new infrastructure that supports a whole research community. CI grants included (a) planning for new CISE community research infrastructure (CI-P grant), (b) the creation of new CISE research infrastructure (CI-New grant), or (c) the enhancement of existing CISE infrastructure (CI-EN grant).

2.2.2. Proposal to CRI Program

I identified the CI portion of the CRI program as an appropriate funding vehicle to potentially address the infrastructure problems related to linguistic annotation. I submitted a planning proposal (CI-P) in late 2013 while a research scientist at MIT, and this small proposal was awarded $100,000 for the period May 1, 2014 to April 20, 2015 (later extended to August 31, 2015 after the grant was moved to FIU). The ultimate goal of the planning proposal was to hold a workshop on the issues at hand, charting a way forward. I proposed 3 tasks in the original proposal:

Task 1: Comprehensive Review. The first task to be carried out, in advance of the actual workshop, was a comprehensive review of the domain of text annotation. The output of this task was to be the identification of the dimensions which define the space of linguistic annotation, including (but not limited to) types of document

[4] http://clarin.eu/

[5] http://www.nsf.gov/pubs/2013/nsf13585/nsf13585.htm

collections, workflows, and annotation types. The plan was to also produce relatively comprehensive lists of corpora, tools, and annotation schema.

The dimensions of linguistic annotation, indexed with concrete examples, were to allow me to identify what types of annotation projects are supported by extant tools. It was also to allow me to specify precisely what types of annotation projects are most critical to the field. This information would, finally, allow me to prioritize implementation decisions, while keeping the grand vision of capabilities in mind.

This review was also to serve as a seed for the workshop, providing a list of names to mine for participants to invite, as well as the raw material that would form the basis of the workshop discussions, allowing us to situate them in context.

In addition to reviewing the published work on annotation tools, I proposed to create a relatively comprehensive repository of extant annotation tools. I proposed to actually run the tools so as to evaluate them directly as to their ease of use, and to mine them for desirable features. Having such a repository in hand, I imagined it would be a short step to actually providing these tools for download from a single source, which is not currently available. Unfortunately, this turned out to be an infeasible amount of work for the time budgeted.

The review was not to be limited only to reviewing published work, or only to tools downloadable from the internet. I also proposed to mine insights and ideas from other researchers, by first sending personal emails to researchers I know to ask them to identify the key tools, corpora, and collaborators for their own work; and then introducing myself to these identified collaborators directly to repeat the inquiries. I proposed to attempt to segment and cluster the field into areas of interest, each potentially requiring different sets of features and capabilities. The goal of this exercise was to produce a comprehensive view of what the community needs, but also have on hand a large body of researchers that we can draw on to make the workshop (and any later work) a success.

While each of these goals was noble and worthy of pursuit, the task turned out to be much too large for the manpower and time budgeted. A survey of linguistic annotation tooling resulted in a bibliography covering approximately 120 tools, formats, and frameworks, containing over 220 articles. It was clear that this represented only a small fraction (perhaps 10%) of the work related to linguistic annotation, and that significant effort would have to be expended to truly map the space. Nevertheless, the survey did allow me to identify a number of important key players, tools, and approaches and bring together a reasonably representative sample of researchers to the actual workshop.

Task 2: Preliminary Design Work. As originally conceived, the proposal sought to explore the feasibility of designing a "Unified Annotation Workbench" (UAW), which would serve as a "one size fits all" tool for linguistic annotation. As discussed in the next section, discussions with researchers in the field during the course of planning the workshop quickly led to the conclusion that a UAW was infeasible and not desirable. Nevertheless, I envisioned in the original proposal that we would put some effort into preliminary design work, to sketch out a possible solution to the general problem of constructing a UAW. I envisioned that this would first entail an identification of a range of potential implementation technologies, including application frameworks, interchange formats, server architectures, and hardware requirements. After this the goal was to produce a block diagram of a proposed UAW architecture, as well as an implementation timeline for development, including when certain features will be available. I also envisioned the creation of initial rough mock-ups of UAW user interfaces, websites, and administrative screens, to give people a sense of what the tool would look and feel like. None of this work was carried out because the premise of the task—a need for a UAW—was eliminated early in the project.

Task 3: Workshop. The highlight of the planning grant was to be a workshop to engage the relevant communities, this technical report describes in detail the workshop that was held. The goal of the workshop was to define the problems, identify the solutions and their benefits, produce recommendations, and get buy-in from key players for future work.

While reviewers of the proposal expressed skepticism that a unified approach would be feasible, they noted that the problems described were real and urgent, and that some sort of community infrastructure solution was needed. It was on this basis the proposal was awarded, and thus the survey work and workshop planning was begun.

3. Goals

3.1. Original Goals

As already noted, the original goal of the proposal for the workshop was to lay the groundwork for a Unified Annotation Workbench (UAW). The vision of this workbench—conceived as a closely-federated set of tools—was to be a one-size-fits-all solution to linguistic annotation, where users could quickly download and install the tool, and quickly visualize, manipulate, and create linguistically annotated data.

When the proposal was funded, I turned to surveying the state of the field, as proposed in Task 1, and also started organizing the workshop, identifying key players in the linguistic annotation space, for example, those who had designed annotation tools, built annotated corpora, or done other significant work on the theory or practice of linguistic annotation. In my discussions with these potential participants, they nearly universally rejected the idea that a one-size-fits-all tool was feasible, possible, or even desirable. The arguments against a UAW could be grouped into to approximately three objections:

First, a number of researchers noted that there is a fundamental trade-off between generality of the tool and its efficiency for a specific annotation task. A tool specifically designed to support a particular annotation task, with a particular type of source material, specific annotation scheme, and particular linguistic judgments, has a huge advantage over more general tools in terms of speed, efficiency, and effectiveness. On the other hand, specific tools were less able to be adapted to new tasks. Projects needed the ability to trade off generality and specificity in their tool choices.

Second, each annotation project has its own set of practical and theoretical constraints which restrict the kinds of tooling that can be used. For example, a project which requires annotators to produce annotations in a low- or no-network environment will not be able to use a tool that requires network connectivity. Similar restrictions apply for particular choices of operating system or computing platform. Some projects have legal or ethical restrictions on how data must be managed or protected, which restricts the kinds of tooling that can be used.

Third, as a matter of efficiency and cost, many noted that a lot of good tools already existed, and while many fell short in certain ways, it certainly made more sense to build on top of those tools rather than tear them down and replace them with a fully new solution. There is a certain amount of accumulated wisdom and practical elimination of bugs and poor practices represented by a vetted tool that is hard to capture in documentation or any one individual's understanding of a tool, and this hard-won quality would be sacrificed in a new implementation.

3.2. Revised Goals & Questions

Because I realized early on that a workshop focused on building a "Unified Annotation Workbench" was ill-advised and not what was needed by the field, I quickly set about revising the goals for the project such that they took into account these objections. I noted that the original motivating problems were still valid, namely, that the existing tool landscape was "balkanized," with a lack of functionality, interoperability, and reusability that were blocking progress. Thus the goals for the workshop reformulated in a more general fashion which did not assume the solution, but rather allowed the participants to work on defining the problem and the appropriate approach to solving it.

Therefore the workshop goal was reformulated as "trying to chart our way forward out of the current chaos of annotation tooling," with the sub-goals of (1) agreeing on the problems with current tool landscape, (2) identifying possible benefits of solving some or all of the problems, (3) identifying which capabilities are most needed and by whom, (4) making high-level suggestions on how these capabilities should be implemented, and (5) discussing how we should go about ensuring longevity and sustainability of the implementation.

Thus the actual agenda of the workshop was driven by five major questions:

1. What are the problems with current tool landscape?
2. What are the possible benefits of solving some or all of these problems?
3. What capabilities are most needed?
4. How should we go about implementing them?
5. How should we manage our solution?

In accordance with this revised approach, I renamed the workshop to the "Workshop on Unified Annotation Tooling", to express our desire to achieve rational annotation tooling, without prescribing a particular solution.

4. Logistics

The next section contains a point-by-point summary of the workshop discussion. Before that detailed explication, however, the reader may find it informative to understand the logistics of the workshop, namely, where the workshop was held, how it was organized, and how the particular participants were identified.

First, the workshop was held on its own in a relatively secluded location: Sunny Isles Beach, Florida. This is not near any university (it is about 45 minutes from FIU), and indeed is not near any tourist sites in the southern Florida area. It is a peaceful stretch of hotels and shops next to the beach. I had considered the possibility of holding the workshop in conjunction with a main conference such as the Association for Computational Linguistics (ACL), the

North American ACL conference (NAACL), or possibly the Conference on Intelligent Text Processing and Computational Linguistics (CICLing).[6] Co-location with a main conference would have had the advantage of reducing travel costs because many of the participants would have already been present. However, I was concerned with the possibility that workshop participants would be distracted by other pressing duties related to the main conference and thus, because the funding afforded by the NSF allowed paying for participant's travel, I decided to hold the workshop in a remote location to minimize distraction and maximize the amount of time and attention participants would pay the topic at hand. Southern Florida is a beautiful place in March, and it seemed reasonable to have the workshop nearby for practical organization purposes, as well as being an enticing place for participants to attend. I chose Sunny Isles because there are a multitude of affordable options for a small meeting, allowing us to have a beach-side venue that I hoped would help participants relax and focus on the topic.

My choice of participants was governed by several principles. **First**, I began with premise that anyone who attended should have some demonstrated accomplishment in linguistic annotation: either having created a significant tool or corpus, run a significant annotation project, or otherwise contributed notably to the practice or theory of linguistic annotation. I generated a list of potential participants that met these criteria from the survey described in the proposal Task 2, and as I discussed the possibility of attending with various potential participants, querying them as to their availability and interest in the topic. I also asked each of them to suggest anyone whom they thought particularly suited to participation. **Second**, it was important to have significant international participation, especially from Europe which has produced the majority of widely-used annotation tools. On the other hand, the NSF-funded nature of the workshop required that U.S. participants be in the majority, because although we would expect that any infrastructure so planned would be globally useful, the NSF is by necessity focused on U.S. researchers. **Third**, while text annotation was primary (as it represents the largest share of linguistic annotation done in the field so far), there should be a diversity of representation of researchers who do linguistic annotation in other modalities, such as speech, gesture, video, multi-modal, and so forth. This would allow the workshop conclusions to be more general and widely-applicable than if the participants were only concerned with text annotation.

In the course of identifying potential participants I held face-to-face meetings, video calls, voice calls, and email discussions with 40-50 researchers in linguistic annotation. I extended approximately 30 rolling invitations. Some invitees developed scheduling conflicts and ended up not coming. The final list of participants and their biographies can be found in Appendices A and B. A photo of the participants is shown in Figure 1.

In the weeks before arriving at the workshop itself, I surveyed the participants. The exact survey is shown in Appendix C, and the full collated (unattributed) list of survey answers is given in Appendix D. This collection was used as raw material to seed the discussion, and was handed out in this exact form to the participants at the beginning of the workshop. The original agenda of the workshop is shown in Appendix E, and was designed to be primarily group discussion, anchored by two presentations. The first presentation was my own context-setting presentation. The second presentation, on the morning of the second day, was by Erhard and Marie Hinrichs and was intended to introduce those unfamiliar with the major European initiative in this area, CLARIN. During the first session of the workshop on Monday morning I asked the participants to suggest reworkings of the schedule. As discussed below in the summary, this resulted in adjustment of the agenda and the addition of breakout sessions as shown in the summary.

[6] The Language Resource and Evaluation Conference (LREC) would have been ideal in this regard, but it is only held every two years and was not scheduled to be held in 2015.

Figure 1: Photograph of the Participants. **Back Row (L to R):** M. Kipp (dark blue shirt); H. Sloetjes; P. Stenetorp; T. Hanke; J. Pustejovsky; E. Nyberg; **Middle Row:** B. MacWhinney (red shirt); E. Hinrichs (partially obscured); M. Verhagen; S. Strassel; M. Dickinson; G. Simons; B. South (partially obscured); J. Good; **Front Row:** A. Rumshisky; W. Chapman; G. Petasis; N. Ide; D. Maynard; M. Hinrichs; C. Bonial; **Kneeling:** M. Finlayson (organizer)

5. Discussion

This section contains a detailed, point-by-point explanation of the discussion as it unfolded, as captured by audio recordings. I transcribed these recordings and then extracted all relevant information into the summary below.

5.1. Summary of Survey Answers

Before the workshop began, I distributed a survey to the participants to gather raw data to guide the discussion. The survey was invaluable in collecting unbiased ideas from all participants before the workshop began and before discussion had resulted in any convergence of views. The exact survey distributed is shown in Appendix C. I analyzed the answers using a clustering exercise, where for each broad question I grouped the answers into coherent categories. Each of these categories of answers is given below in Table 1 (in alphabetical order), and the detailed list of collated answers is given in Appendix E. These categories allowed us to think carefully about what was missing in our approach to the problem. The full collated list was distributed to the participants the night before the workshop began and was available throughout to seed and guide the discussion.

5.2. Summary of Workshop Discussion

5.2.1. Monday, Session 1: Context, Agenda, Problems

I began the workshop with a presentation that reviewed our goals and the context under which I organized the workshop. The goals, as stated previously, revolved around addressing the problem of the balkanization of linguistic annotation tooling. The context covered the grant from NSF CISE Research Infrastructure program.

With regard to the goals I emphasized that we were not discussing "yet another annotation tool." In the surveys and pre-workshop calls participants noted, with minor qualifications, that "the solution to the balkanization of annotation tooling is not to build another tool."[7] I presented this in a generalized form for the overall goal of the workshop, namely, "we are trying to chart our way forward out of the current chaos of annotation tooling."

Next in my presentation I reviewed the results of the pre-workshop survey, which are outlined above in §5.1, and listed in detail in Appendix D. Importantly, there were two points on which there was substantial agreement, in that these two points were made in the survey by nearly every participant in one form or another:

1. Any solution should build links and reuse existing tools as much as possible, and
2. Participants universally desired better capabilities to control one's annotation workflow.

What are the problems with existing tools?

- Difficult to learn
- Difficulty running or installing
- Inadequate importing, exporting, or conversion
- Lack of documentation or support
- Missing functionality
- Poor user interface
- Problems with extensibility
- Unstable, slow, or buggy

How should we implement a solution?

- Build links and reuse existing tools
- Tools are difficult to reuse
- No one tool is possible
- Miscellaneous

What capabilities are desired?

- Import/Export capabilities
- Annotation on various media
- Supported schemes
- Workflows
- Miscellaneous

What are our motivations to solve the problem?

- Achieve greater access to data
- Avoid reduplication and reduce cost
- Ease annotation
- Facilitate archiving
- Improve the range of annotations
- Increase quality of results
- Obtain interoperability
- Provide extensibility

What disciplines should be prioritized?

- No prioritization
- Some prioritization needed

How should we organize, fund, and manage the solution?

- Funding
- Implementation
- Open Source
- Organization
- Partnerships
- Review

Table 1: Broad categories of answers to core questions, as extracted from the survey data.

After the presentation, the first item for discussion was the agenda of the workshop itself, and whether there were any suggestions for modifying the proposed order of discussion, including whether we have a complete list of problems from which to start. This resulted in the following observations and suggestions:

1. The surveys indicated little interest in prioritizing disciplines and so I dropped that from the agenda. (As it happens this was discussed later under the guise of "audiences," in §5.2.3). In particular, one participant noted that disciplinary distinctions are not as useful as the question of "what is the role and impact of language in a particular annotation task?"
2. One participant observed that we should make sure to discuss things that had already been tried in solving the problem of balkanization, so that we may avoid repeating previous mistakes.
3. Several participants suggested that we discuss use cases to motivate capabilities, and the idea of breakout groups was suggested to generate use cases focused on particular problems.
4. One participant forwarded the idea of creating a classification scheme for annotation tasks and tool capabilities. This was also offered as a way of organizing potential breakout groups.
5. Another participant suggested the idea of creating a roadmap, to lay out disciplines, tasks, and use cases so we know both where we are and what direction the field is headed.

[7] Within this summary section (§5.2), quotes are drawn directly from the transcript of the discussion, but are sometimes left unattributed at the request of the workshop participants.

We also treated in more detail the overall purpose of the workshop. It was reemphasized that we need to go beyond individual tools and corpora, and that one of the most important questions relating to balkanization is how do we encourage and enable *semantic* interoperability between tools. This was situated in a three-level understanding:

1. **Syntactic**: The "file format," which lays out how the data model is written to disk.
2. **Logical**: The "data model," which lays out the logical organization of the annotations and how it maps to the conceptual model and annotation scheme. This is also often called the *annotation specification*.
3. **Semantic**: The "annotation scheme," which is covered in the annotation guide and finds its fullest expression in the conceptual models held in the minds of the annotators and investigators.

Participants emphasized that the main concern is not at the level of syntactic interoperability, namely, making sure that file formats are the same: there has already been quite a bit of work on this in terms of LAF, GrAF, MASC, and so forth [16], [17]. Logical interoperability, furthermore, was also fairly well treated, in that type systems like that in UIMA and GrAF had made inroads on this problem. Rather, existing linguistic annotation tooling did not enable *semantic interoperability*, and this lack is a critical barrier to effectively reusing annotation schemes, annotated corpora, and linguistic annotation tools. Solving this problem would also enable and assist tool writers for the next-generation of tools.

It was in this discussion that the idea of central repository first started to take shape. The idea was to start first by capturing the state of the field, organizing a central repository that lists all available tools and annotated data, how they interoperate, and how different data formats can be converted between.

One participant, however, sounded a note of caution on focusing only on building a central repository. They noted that the current funding climate was heavily biased to "visionary" research that takes risks and promises large rewards. They noted that to be visionary you need a visionary use case, which led to the idea of outlining a major challenge problem that can motivate the infrastructure. They noted that it is better to shoot for something that the NSF knows no company can do. Other participants, however, countered that really high-risk and visionary challenge tasks are more suited to other types of funding, such as DoD and DARPA.

At this point I turned the discussion back to considering the problems we are having as a field. I asked again whether the list of problems generated from the survey was complete.

After some quiet thought and consideration of the existing list of problems, one participant noted that "discovery" is a major problem category not represented, where discovery was defined as "the problem is there is a tool out there, but I don't what it is, and I don't know where it is." Thus the problem of *discovery* is the difficulty in finding the tools that one needs. The participant noted that there are "actually some very simplistic infrastructure solutions" that can address this need.

This observation returned the discussion to the idea of a central repository for tools and data. Some noted that there had been efforts to catalog tools, but others countered that these efforts have fallen short. A number of out-of-date or incomplete webpages were mentioned, such as those at LDC[8] or in Germany[9]. OLAC[10] was mentioned in this context, because it includes some tools and is, generally, a catalog for language-related things. Participants admitted, however, that OLAC did not solve the problem at hand, and any tools it included were included in an *ad hoc* and inconsistent manner. The out-of-date nature of prior efforts led participants to a short discussion of the problem of sustainability, namely, keeping any catalog up to date and current requires significant continuing effort.

At this point one participant proposed the idea of adding formal metadata describing tools. They argued, and others agreed, that formal metadata over tools would add major value to the repository, and would be something no repository (incomplete or not) has done before. The participants connected this idea of formal metadata back to the idea of enabling semantic interoperability, with the metadata ideally being specific and precise enough to capture how and in what way tools and annotation schemes were semantically interoperable.

As the end of the session approached, I asked for final brainstorming on problems that should be discussed. The additional problem categories suggested were:

1. No support for annotating in low-resource environments, such as where you have no network access
2. Lack of workflow control and manipulation

[8] https://www.ldc.upenn.edu/language-resources/tools
[9] http://annotation.exmaralda.org
[10] http://www.language-archives.org/

3. No clear direction for evolving tools for the next generation
4. Lack of support for developing annotation guidelines in a formal way
5. Difficulty managing large volumes of data or annotators, in particular, crowd-sourcing
6. No support for embedding annotation in other tasks (e.g., Games with a Purpose, or GWAP)
7. Limited ability to support human-in-the-loop for annotation

5.2.2. Monday, Session 2: Prioritizing the Problems

After the coffee break the discussion resumed, beginning with a round of introductions. The goal of this session was to prioritize the problem categories identified in the survey and augmented in the first session. As each problem was prioritized, it was discussed and often reworded. In the end, with various merging and splitting, ten problem categories emerged, organized into three priority levels (high, medium, low). Within the categories the priority is unordered. The summary below of the discussion is organized according the final prioritized list.

High-priority Problems

There were five high-priority problems, and they are listed here in no particular order.

Difficulty finding and evaluating appropriate tools and resources for the task at hand. One high-priority problem was simple: the difficulty people have in merely finding tools and data for their particular tasks. This was described earlier as the problem of *discovery*. This is bound up with the difficulty of evaluating the appropriateness of a tool for the job: once you have found a tool, it is hard to know if it will work for you. One participant noted that "you have to get pretty deep into a tool to know if it's going to work for you, and knowing that, people, including us, often opt to just build their own tool out of the box, because of the time investment to evaluate the appropriateness of a tool" is significant. Participants discussed how they approach this problem now, and several suggested that word of mouth is important, namely, asking other researchers "have you used it, and how did it work for you?" Continuing with the theme of a central repository, it was noted that some sort of standard vocabulary would be incredibly useful: if all tool providers that register their tools in the repository use the same set of concepts to describe their tools, this could be quite helpful.

Lack of support for development, documentation, reuse, naming, and formalization of annotation schemes and guidelines. This problem revolves around support for formal ways of describing the semantics and interoperability of annotation schemes. Annotation schemes and guidelines have a lifecycle that runs from development, to testing, to documentation, to reuse, and the ability to formally name and describe the contents of annotation schemes and their related guidelines would be a significant step forward. Analogies were made with the utility of, for example, three-letter ISO scheme for naming languages, or the notion of a MIME type. Other related tasks that such formalization that are now difficult to achieve are versioning of annotation schemes, customizing existing annotation schemes, standardizing approaches, and merging disparate approaches to annotation. In the context of documentation, one participant mentioned that Pandoc[11] was an exemplar project and could be considered as a model.

Inadequate Importing, Exporting, Conversion. With lots of different tools have come lots of different formats, with varying compatibility across different levels (syntactic, logical, and semantic). While semantic interoperability has already been noted as a separate challenge, there are interoperability problems even when annotation schemes are logically compatible. A missing piece for the field is that the ability to easily import into one's tool of choice logically compatible annotations stored in different syntactic formats. It was suggested that a stand-alone service that can convert data from one format into any other compatible format would be of great use in this context.

Lack of Workflow Support. Workflow is the overall structure of the work of an annotation project, including chaining tasks, pipelining tools, and assuring quality. Right now, there is little established workflow support, and we have no ability to describe workflows in uniform terms. Some participants noted that there are existing solutions for workflow management, for example, Taverna[12], but that these were either not used in linguistic annotation or not appropriate for one reason or another. The problem of workflow support has several different aspects. **First**, one of the critical difficulties is in pipelining tasks, especially across several annotation tools. Most annotation tools do not naturally speak to each other, and so the addition of each tool to a workflow requires additional steps for importing

[11] http://pandoc.org/
[12] http://www.taverna.org.uk/

and exporting data, with consequent possibilities for errors and problems. A specialized case of this is when a human is introduced into the annotation pipeline, as they often are: despite the centrality of human annotators, there is little formal support for human-in-the-loop in a controlled way. **Second**, there are no unified mechanisms for ensuring quality across human and machine annotations. There has been some research on checking for and finding errors in annotated data [18], but these approaches are not yet generally used. A proper workflow solution would enable and encourage quality assurance steps. **Third**, workflows developed for one task, if they are clearly described and portable at all, are not easily adapted to different, but related, tasks. This generally is a problem of extensibility: if workflow solutions are not extensible, then this significantly impairs their reusability. **Finally**, lack of workflow support results in major difficulties for non-technical researchers (e.g., Digital Humanists, Sociolinguists) when they want to do linguistic annotation: they lack the significant technical knowledge required to chain together tools in a way that accomplishes their goals.

Lack of User, Project, and Data Management. A problem that cuts across all types of tools is the inability to scale to large volumes of annotators or data. Existing tools are particularly unsuited to crowd-sourcing, which is becoming more and more popular. This was considered by a number of participants as the number one problem affecting many annotation projects. With regard to data, there were several specific problems. First, **searching** data—especially multi-modal data—is difficult, meaning it is difficult to find pieces of data in your raw source or annotations that meet certain criteria. Currently any search system must be custom made. Second, **versioning** is a problem: it is difficult to inspect the history of changes applied to one's data, and it is difficult to roll back to previous versions or to merge two different versions from different annotators into a new version (or gold standard). Code versioning systems (such as SVN or git) were suggested as potential solutions or models, but it was clear that the field needs capabilities specific to linguistic annotation.

Workflow support and user, project, and data management are clearly related, and to conceptualize the relationship participants generated a multi-level arrangement:

1. **Task Intention**: defining the workflow from the point of view of the requirements of the analysis
2. **Logical Parts**: breaking the intention down into logical subtasks and identifying who will perform what
3. **Implementation:** designing how to implement the subtasks with tools, selecting and provisioning the actual resources, and then effecting execution.

As with workflow support, extensibility of any eventual solution was seen as a necessary feature to facilitate reuse.

Medium-priority Problems

There were two medium-priority problems, listed here in no particular order.

Learning Curve Too Difficult. Linguistic annotation tools are almost uniformly hard to use and hard to learn how to use. While a tool may in theory be functional in a particular way, it is often too difficult to learn how to use it to actually achieve that functionality, and using it to achieve that functionality is clunky and difficult. One participant phrased this as answer the questions, "how do you download this tool, how do you install it, how do you run it, what do all the buttons do?"

This problem is a function of several related sub-problems, in particular, the design of the user interface or the existence of comprehensive, up-to-date documentation. Importantly, for many tools, the learning curve is too shallow, primarily because the user interface is really clunky: more investment of effort in using and learning the tool does not result in major annotation performance gains. This problem also results in some people who should be using a tool not being able to use that tool, such as non-technical researchers interested in linguistic annotation.

There was quite a bit of vigorous debate about whether to put this problem in the "high" or "medium" categories. Some participants thought the learning curve was a serious problem (see the example below), while others saw it as more a software engineering problem that was a sad fact of academic life, that either would be solved through application of known techniques or was not our job to solve (see low-priority problems, below).

Stephanie Strassel offered a specific example to support putting the problem in the high-priority category. The Alembic workbench is an older but still very powerful and flexible annotation tool in use by the Linguistic Data Consortium (LDC). Unfortunately, the power and flexibility of Alembic also means it is easy to do an annotation incorrectly, and difficult to learn to do it right. Even for technically minded people it's hard to learn, and for non-technically minded people (annotators), it can be a major blocker. From the LDC's perspective as a large corpus developer, they may have a team of 50 annotators distributed around the world, and they have difficulty finding

people who speak the language being annotated, let alone also have strong computer skills. In this case, the difficulty of the learning curve is a serious concern.

Discussion of this problem led to two interesting suggestions. **First**, one participant suggested that a series of usability studies could be useful, with different target groups (e.g., annotators, project manager, researchers), to try to understand the problem of usability of linguistic annotation tools. They noted that the whole question of what is a usable tool is not a simple question, and that they are likely separable questions for different populations: there's a learning curve for the annotator, and there's a learning curve for the researchers, etc. Others noted that usability also was related to the tension between theory-specific and generic tools, with specific tools being more difficult to learn, but more efficient in the long run, while generic tools are easier to learn but less useful for specific tasks.

The **second** suggestion related to the lack of community-wide UI or task idioms for linguistic annotation. Jeff Good noted the following:

> *Elan [the annotation tool] might call something a "symbolic subdivision". I know ELAN's word for that. But then I go to another tool and know that I need a symbolic subdivision, but I'll spend half an hour to find what they call a symbolic subdivision. … That's part of the problem. You already know what tasks you want, you just don't know what they're called. If you look at Word processors, right you have File and Edit, which is what allows the learning curve. The first word processor is [difficult], but second one is easier. There is a lack of a community-wide idiom.*

A community-wide idiom for annotation tooling may help ease knowledge transfer between applications.

Lack of Tool-building Tools. Another much discussed and desired capability, but one that was ultimately downgraded to "medium" priority, was a set of tools that would allow researchers to quickly and easily build functional, extensible annotation tools with good user interfaces. This problem was downgraded in the end because participants felt that certain parts of this problem would be solved (especially with regard to user interface), when we make progress on the high priority items, especially those related to managing workflow.

The specific problem was conceived to be the inability to perform a number of common software design tasks using pre-built components. For example, one participant proposed the idea of a wire-framing interface to design the layout that integrates existing UI components. Another participant regretted our inability to easily build different types of interfaces or visualizations based on the annotation scheme. These ideas lead naturally to the idea of plug-and-play components: for example, if a researcher needs to visualize trees, or another type of annotation data structure, there should be components that you can drop into the application to perform these functions.

This capability, the participants felt, would naturally lead to better user interfaces, would allow tools to be easily retrofitted with missing functionality (including annotation schemes), and would address problems with extensibility. But it was also noted that having plug-and-play tool-building tools assumes that you have solved some of the higher-level problems, such as data management, scheme design, document interoperability, as well as understood more precisely how to manage your annotation workflow. Therefore, participants agreed to mark this as medium priority, whose solution would be partially dependent on higher-priority issues.

Low-priority Problems

There were three low-priority problems, listed here in no particular order.

Inability to do Opportunistic Annotation. One problem that was suggested was that we have difficulty making annotation "fun", or, more generally, we lack the ability to embed annotation opportunistically in other non-annotation tasks. This is related to the idea of "Games with a Purpose" (GWAP), where there has been significant recent work and several examples within the linguistic annotation space.

Another related problem that one participant brought up was that it is difficult to do purely opportunistic annotation as a researcher: for example, if you are reading some text and note an interesting linguistic fact, there is no uniform way to reliably and reproducibly annotate them. The best you can do, right now, is to keep notes.

While participants agreed that both of these problems were interesting and solving them could be useful, the general feeling was that this was perhaps somewhat on the edge of the scope of the central problems of the workshop. This problem was thus relegated to the "low-priority" category.

Difficulty Running or Installing. A second low-priority problem, although a common one that frustrates many researchers, is the difficulty in merely installing and running existing annotation tools. There may be an annotation

tool that provides some needed functionality, but it may not work on a researcher's platform of choice (which includes, for example, the distinction between desktop and mobile). While participants noted this was a frustrating problem, they believed that solving the higher-level problem of workflow management would go a long way toward solving this problem. Further, participants noted that this was a specific, pernicious example of a more general class of problems revolving around poor software engineering, which was placed in the low-priority category for reasons explained next.

Poor Software Engineering. By extension with the previous problem, many existing annotation tools suffer from poor software engineering at one or more levels: tools may be unstable, slow, or buggy; they may lack documentation or support from their developer; and the code may be difficult to understand or use, if it is available at all. While participants all agreed that these were all serious problems, they were not unique to annotation tooling but rather were the result of academic code as a whole, and generally were a function of the incentives of the academic world (jobs and promotions being awarded on the basis of publications, and not cleanly written code). It was admitted that, no matter how frustrating the low quality of software and the lack of general software engineering hygiene, this was just not something we could tackle at the level of the field, and it is not really an infrastructure question. This problem was thus placed in the "low-priority" category.

Orphaned Ideas

There were two interesting ideas that came out of the problem prioritization discussion, and which did not fit cleanly in other categories. I list them here as two "orphaned" ideas.

Annotation Skill Ecosystem: One participant suggested that it could be quite useful to create an annotation task ecosystem, where specific skill sets related to annotation could be advertised and recruited, like Amazon Mechanical Turk. This idea developed as an extension of the idea of how to manage interoperability between schemes and data: if one talks about independent sources of data and schemes, why not independent sources of task management and even code? A specific example was offered of this service: Suppose there is a person in the U.S. who is a great annotation task manager and would like to work as a freelancer in their spare time. A researcher in Germany has a bunch of students who can do annotation, but doesn't know how to manage the annotation task. An ecosystem portal could potentially bring these two together. While this idea was interesting and exciting it was unclear to the participants that the lack of such a capability was a blocking problem, and in the end it was decided that this was somewhat out of scope for the workshop.

Data Repository: Another interesting idea was the idea of building a generic data repository for annotated data. Such a data repository would be accessed via a simple API which supports several straightforward calls. This would then allow annotated data to be moved off the local machine in many cases and also allow annotations from many different sources around the world to be brought together. It was suggested that it could be a repository only in a very general sense, like OLAC, and on the backend be distributed. Importantly, however, to the end user it should look like one thing, so there is one single point of entry. While this idea was intriguing, and was eventually integrated into the final recommendations, it was unclear to the participants if this was a blocking problem, and therefore the lack of such a facility was not promoted to the status of an independent "problem."

5.2.3. Monday, Session 2: Identifying the Audience

During the discussion to prioritize the problems the participants noted that to place problems in their appropriate category (high, medium, low), we needed to know who are target audience was, that is, who is the population our solutions are designed to serve? Some problems were higher priority for different audiences: e.g., user interface problems should be higher priority if our audience is the annotators themselves, while workflow management should be higher priority if the audience is project managers. This led to an extended sidebar where we discussed the appropriate target audience for the solutions proposed at the workshop.

We started by listing possible audiences, and they included (in no particular order):

- **Social Scientists:** Social scientists do a lot of annotation-like activities, often supporting measurements like content analysis [19]. Further, social scientists are often non-technical and have great difficulty using extant annotation tooling.
- **Digital Humanists:** Even more than social scientists, digital humanists use a lot of annotation in their work, and it is almost exclusively language data. They have less severe problems with lack of technical expertise, but this is still a challenge for them.

- **Computer Scientists:** This category was meant to include computationally savvy computational researchers (who are not computational linguists) who use language data in their work.
- **Computational Linguists:** While there is a thriving subfield of computational linguistics dealing with the production of annotated data, most computational linguists merely use the annotated data and potentially would benefit from being able to more easily generate the annotations themselves.
- **Annotation Tool Developers:** Cutting across different fields, annotation tool developers are those who actually build the annotation tools used by other researchers. These developers may or may not actually have annotation projects of their own (it is often the case that they do).
- **Annotation Project Managers:** These are skilled workers—often graduate students, but also perhaps paid employees—who actually handle the day-to-day details of running an annotation project: they recruit and train annotators, manage the data, check the results, and ensure that the gold standard is properly constructed.
- **Annotators:** These are the trained workers who apply annotations to text. Their overriding common feature is knowledge of the language being annotated. They may be trained or untrained, computer savvy or not, and are the ones who spend the most time in front of an annotation tool UI.
- **Linguists:** This category is meant to include non-computational linguistics, including field linguists, grammarians, historical linguistics, comparative linguists, and so forth. These researchers have a great need to create and use linguistic annotations, but are usually not computationally savvy enough to use the extant tools. Importantly, they do tend to share a common idiom for linguistic ideas, which makes them a much more unified audience.
- **Speech & Hearing Science Researchers:** These are researchers who specifically focus on speech and hearing, and would benefit from significant improvement in the tools they use to produce and manage their annotations.
- **Other Researchers:** This category was meant to capture anyone who has a really clear idea of what they want to do with regard to linguistic annotation, whether they are computationally savvy or not, and who really needs help to manage the tooling.

It was first noted that among these classes of audience, there was a major two-way sub-division: (1) On the one hand, there are researchers whose only experience with computers is through commodity end-user applications. These include humanities or social science researchers, where to be successful the annotation tools really need to be as easy to use as Microsoft Word or Excel; (2) On the other hand are researchers who are able to program and develop their own tools to a certain degree. In this case you want more of a sandbox or toolkit solution, where the developer can make their own tools. By way of example of this distinction, Steve Cassidy mentioned the example of the Galaxy platform [20]. Galaxy is a platform for standardization, reuse, and sharing of biological data, resources, and tools. Its uses for researchers are similar to the developers that work on natural language processing, computational linguistics, corpora development, and so forth. Galaxy is a separate community similar to linguistic institutions like LDC and ELRA. Galaxy's main goal is to wrap tools in ways that enable everyone to use them. The paradigm as described allows a division of labor between (1) technically proficient researchers who write extremely functional but difficult to use tools, (2) somewhat technically proficient researchers who are able to understand those tool well enough to write a Galaxy wrapper for it, and (3) other researchers who only use the tools through the wrappers.

After some discussion, the participants were able to come up with a four-way prioritized abstraction into which each of the above types of people could be placed:

1. **Specialized Corpus Creators:** This includes annotation task managers and tool developers. These researchers are experienced in linguistic annotation, can conceive an annotation task, know what tools need to be built, hire and manage annotators, and so forth.
2. **Non-specialized Corpus Creators:** This category includes those for whom building a corpus is a secondary concern to their primary work. This includes, for example, theoretical linguists, social scientists, and digital humanists to a certain degree
3. **Annotators:** The people who actually apply annotations to language artifacts.
4. **Annotation Users:** Researchers who use annotations to do science.

The above list is prioritized, in that the participants emphasized that the point of the workshop and the project at hand was to provide infrastructure to help execute annotation projects. The participants noted that if specialized corpus creators are not well supported in their task, then the other categories of people have little hope of finding

relief for their problems. The sentiment was that if those with deep expertise in annotation and annotation tool design cannot accomplish their work and do annotation, then everyone else with less technical proficiency will have a harder time. Thus, the priority target audience was identified as "**corpus creators, annotation task managers, and developers**"

Once this decision had been made, we went back and checked that all the problems and their priorities settled on prior to that discussion still made sense in the context of this more specific audience focus. This concluded the first morning of the workshop, and the participants broke for lunch.

5.2.4. Monday, Sessions 3 & 4: Breakout Groups

After lunch, and in accordance with suggestions made in the first session, our goal was to split into breakout groups and try to come up with motivating use cases for the high-priority problems that were identified in the previous session. We spent significant time discussing how to organize the breakout groups, and there were at least three proposals for aligning the breakout groups (listed here in order of discussion):

1. Align the breakout groups directly with the high-priority problems identified in the last session. The goal for each group would be to come up with a high-priority use case which is blocked by that problem.
2. Align the breakouts with a preliminary classification of kinds of annotation tasks. The classification dimensions proposed were based on a presentation by Rumshisky & Pustejovsky[13] and included:
    a. Entities vs. Relations: Annotation of entity types and their attributes corresponds to linguistically local annotation, versus annotation of relations that span widely across text and connect entities.
    b. Overt vs. Covert: Annotation that captures elements that are overtly mentioned in the linguistic artifact (e.g., are lexicalized), versus annotation of elements that must be inferred by the annotator.
    c. Span-specific vs. Span-independent: Annotations that are anchored to specific spans or regions in a linguistic artifact, versus annotations that are independent of any particular region or span.
3. Align the breakouts with target media of annotations. It was proposed that this would be orthogonal to the last proposal.

    Ultimately the group decided on the last alignment, along the dimension of media targets, and we conducted an exercise where each participant suggested two corpora that could be placed in one of categories. The final types of media included:

    a. **Text:** Annotation over characters, including transcriptions of speech or other audio, or annotating translations. Examples: PropBank, BioNLP, NUCL, FactBank, EuroParl, BNC, GNC, OntoNotes, TüBa-D/Z, Shared Clinical Reports, i2b2, ACE, MASC, OANC, TimeBank.
    b. **Speech:** Annotation on the audio stream itself, like phonetic annotation. This sort of annotation is distinguished from text because it is time-aligned rather than character-aligned. Examples: AusTalk, CHILDES.
    c. **Multimodal:** Annotation on video or images, or anything that includes both text and non-text data (e.g., audio, images, video). Examples: CNGT, YoutubeParaphrases, TrecVID, Mumin, Gemep, TUNA, Madcat, Confluence Project.
    d. **Metadata:** Annotation of features that are independent of particular spans or regions in the artifacts. An example of this type of annotation would be language types and the types of content in a corpus. Examples: GermaNet, OLAC, DoBeS, SemLink, VLO.

Other media types that were proposed but ultimately rejected were:

1. "Naked media" or "Naked video": This is raw media with no transcription. This was rejected on the premise that these do not contain annotations and so are out of scope.
2. Aligned Translations: While interesting, this was not considered general enough to support a breakout group.
3. Lexical resources: One participant suggested that if we are guided by the principle that annotation is "application of human judgments to language data," then lexical resources are a form of annotation. But the general feeling of the other participants was that, while that principle is reasonable, this broadened the scope of the project in an infeasible way.

Several issues that were identified as being relevant to all the areas included:

[13] http://annotation.co

1. Relations across documents
2. Resource linking

Once the alignment was decided, participants volunteered for various groups, with an eye toward balancing the size of the groups and making sure that experts in the media type in question were present in each group. After the breakout groups met (for approximately an hour and a half), all participants reconvened to summarize the use cases they produced:

Text Breakout Group

This group considered three main categories of use cases blocked by one or more of the high-priority problems: (1) Layered annotation, such as adding named entity detection and coreference on top of existing layers, potentially from a corpus you did not yourself create. (2) Corpus organization, where you have several different levels of analysis, including, say annotations within documents, about documents, or across documents. Here a set of medical records for a single person was offered as a compelling example. (3) Textual generation, which includes tasks like textual entailment or error annotation, where annotators actually generate new text (such as propositions or corrections) that are then annotated. For all three use cases, the group noted that is extremely hard to figure out what tools exist, what tools can be used for the task, what tools are best under what circumstances, and how tools can be hooked together. The group emphatically expressed that *interoperability* was the biggest obstacle to being able to take different tools and put them together in a pipeline or a workflow. This problem extends beyond the first task of being able to "plug-and-play" different tools (and thereby see what works better); it extends to defining the semantics for your own annotations or overall project and judging how well your semantics maps to the semantics of someone else's project. These semantic considerations are critical to whether you can and how you must combine different parts of your annotation workflow.

Speech Breakout Group

This group considered two uses cases. (1) Transcribing overlapping speech, such as dialog or multi-party conversation. The group noted that this relates strongly to the layering use cases identified by the text group. The usual solution for transcribing overlapping speech is to do it in Word, with overlap between speakers indicated by orthographic alignment of two lines. The group noted that tools like ELAN can do this already [21], but then further noted that researchers, especially non-technical researchers, don't use ELAN for two reasons: First, they don't know about them, which speaks to the problem of not being able to find the right tools. Two, the data in ELAN is hard to transform into a publishable format, which is an example of the problem of lack of conversion from the format of the tool to a desired external format, namely, a publication-ready format.

(2) The second use case involved the field linguist's task of collecting recordings and annotating them, with the goal of constructing a full grammar for the language. In this use case, the researcher is developing and debugging the annotation scheme at the same time they are developing and debugging the annotation. Our inability to co-evolve the scheme and the annotations themselves is a blocker here, which is part of the problem of lack of support for annotation scheme and guideline development. Several examples were offered of this problem. Steve Cassidy noted that he had observed researchers working with ELAN, for example, who change their annotation halfway through the annotation of a corpus. In these cases they may, for example, change the tiers they use to annotate the data, but then the annotators forget to—or only incompletely—go back to adjust the previously annotated data. This oversight comes from a lack of validation phase that can confirm that all of the data conforms to some specified scheme. Jeff Good provided another example that could happen even when there is only a single annotator. He noted that in the case of a long-term field worker, they may be generating annotations of an extended period of time (say, twenty years), and their annotation scheme is slowly changing over that time, often implicitly. So in this case, the annotator is working with previous versions of themselves, and must ensure that the scheme at the end applies to all the data generated. That annotator may want to know, for example, that eight years prior they applied a specific analysis to generate annotations but that now, in the present, their theoretical model has changed and they must update their eight-year-old annotations to comply with the new model.

This group emphasized that workflow management is really a critical blocker for non-technical users. When a tool is not a stand-alone application they download onto the computer and double-click to install, this can cause major difficulties for non-technical users. It was also noted that even having detailed instructions does not necessarily solve the problem: the MAUS system [22] from Munich is quite easy to use and clearly described online in a FAQ, but they still get a lot of queries such as, "I want to use it on my data, how do I do it?" One of the fundamental

problems here is that, for non-technical researchers, the learning curve for any new tool—no matter how well documented—is difficult and thus makes for a significant barrier to entry.

Multimodal Breakout Group

This group focused on the so-called ALADDIN[14] use case, which combined human and automatic annotation of a video dataset. The human annotations provide gold-standard data, and the team is simultaneously developing tools to perform automatic annotation across different modalities. One aspect that makes this use case unique is that querying does not happen textually: because the data is multimodal, the only way you can do a query is by example. This requires you to have a sophisticated tool to actually construct queries, which currently does not exist. For multimodal data, it was further noted, the sheer variety of types of anchors used in the raw data is challenging in itself, and for lots of different modalities is hard for a single researcher or even a small team to integrate—or even be aware of—the best-of-breed tool for each of these modalities.

Thus the group noted that this use case was blocked by several major problems. **First**, the core goal of the use case was blocked by the lack of some sort of "meta tool" which allows one to link annotations across modalities. **Second**, the use case is blocked by our inability to find best-of-breed tools easily, and understand how they can work together quickly, which speaks to a lack of interoperability. This is a function, too, of a lack of formal ways of describing annotation guidelines and annotation schemes which allow you to formally specify how the semantics of different schemes map to each other. **Third**, this use case is also blocked by lack of pipelining support and workflow support, which allows one to chain together automatic analysis or annotation tooling, and allows you to quickly run big experiments. **Finally**, because this use case involved video data, bandwidth constraints become critically important for the end user. This led the group to think about how to deal with large data sets, where one must think carefully about the data management facilities and the compute and bandwidth capacities at the annotator's end.

In response to these problems, the group began to brainstorm on potential solutions. One idea that the group floated was having every annotation tool publish minimal constraints and anchors for its annotation layers, using unique identifiers to refer to layer of annotation, which would allow another tool to relate its own annotations to that layer in a loose way. This idea of loose coupling was extended to having tools expose simple functions or configuration information in a standard way. If the community could encourage developers to offer scripting languages (or a set of external function calls or configuration parameters) to allow remote control their tools, this would significantly enhance our ability to create meta tools or workflow management solutions.

Metadata Breakout Group

The final group focused on two different specific use cases: tools that would allow (1) orthography normalization for German; and (2) detection of *vocal fry*. The group noted that if we want to create, publish, and find tools that help with extremely specific use cases (and the vast number of equally specific analogs), the first thing that the field needs is some kind of metadata that distinguishes the types of tools and the types of annotation those tools allow, including additional information such as what import and export formats they support, etc. This led to a discussion of the level of the granularity needed in tool metadata.

According to the group reporter, the primary take away from this group is that the field is missing a (coarse-grained) taxonomy of annotation tasks in which to describe the abilities of tools. Such a taxonomy would allow us to design functional pipelines and work, step by step, down to the micro-level of what import and output formats are supported by what tools. This would allow us to either convert between these different input and output types as allowed, or write appropriate data wrapper that fill the gaps. But this can only be done after we can establish all of the metadata that would be required for a particular tool.

Conclusion of Day 1

After the breakout group report there was a brief discussion about whether or not the schedule for the next day should be altered. The original schedule can be seen in Appendix E, and contained the following order of sessions: (1) introduction to CLARIN; (2) discussion of needed capabilities; (3) discussion of how the solution should be funded, built, and maintained; and (4) summarize the recommendations.

I emphasized that it was important that we map the problems we have identified into capabilities that are needed or wanted, resulting in a relatively concrete list that can be used to focus future work. Again I emphasized that the

[14] http://www.aladdinproject.org/

solution was likely not a single piece of software, but could be any number of things, including websites, catalogs, metadata standards, descriptions of the contents of corpora, and so forth. The participants eventually agreed that we should retain our goal of matching each of the high-priority problems on our list with specific capabilities that might solve them.

The participants also agreed that we should retain a discussion of how any solution would be built, maintained, and managed: namely, the social infrastructure supporting the research infrastructure. One concrete suggestion in this vein was to focus on how to secure funding, how to encourage collaboration, and what future meetings would occur. Eric Nyberg noted, in this context, that it was important to involve anyone from industry who was trying to surmount these specific problems, because those types of people were incredibly helpful when forming the UIMA standards and specifications. He suggested that we may want to perform—either at this workshop or a future meeting—some sort of "roadmapping" exercise. Many participants consider this idea insightful, but all agreed that this was probably too ambitious for a single day, and should be deferred to perhaps its own future meeting.

In the end, the schedule for the second day was retained mostly as proposed, and in practice the discussion naturally expanded or contracted to emphasize the points of most concern to the participants, as reflected in the discussion below.

5.2.5. Tuesday, Session 1: CLARIN

Tuesday began with a presentation led by Erhard Hinrichs, Project Coordinator and Head of the German Section of CLARIN and Member of the CLARIN Executive Board. The goal of his presentation was to orient the non-European participants to this important potential collaborator and stimulate discussion about existing tools that may fulfill needed capabilities.

Up-to date details on CLARIN may be found through their own website[15]. Here I give only a brief summary of the presentation, to retain the flavor (and some context) of what was discussed at the workshop.

CLARIN stands for "Common Language Research and Technology Infrastructure", and is a European funded effort that is intended to support researchers from the social sciences and humanities, where "humanities" is defined as the complement of the hard sciences. The specific goal of the project is to construct and then use an integrated, interoperable, and scalable research infrastructure. An important part of the project is a network of centers, distributed across Europe (and potentially the rest of the world). Many of these CLARIN participants, centers or not, existed previously as data repositories or as providers of tools and corpora (including both spoken and multimodal speech corpora). CLARIN is a way of joining forces and pursuing common goals.

CLARIN is part of the ESFRI (European Strategy Forum on Research Infrastructures) roadmap for research infrastructures. This roadmap comprises three phases: preparation, construction, and exploitation. There are 55 research infrastructures on this roadmap, spanning all the way from medicine and the life sciences to social sciences and the humanities. This roadmap was significant, in that it involved a new pan-European legal framework. Funding for the infrastructures is a mixture of national and European funding. Importantly, an institution can become a CLARIN center even if it is not in Europe: Carnegie Mellon University is currently a CLARIN center, and CLARIN is in negotiations with Tufts University (via the PERSEUS[16] project with Gregory Crane) to become one as well.

What do CLARIN centers do? They provide central services, like persistent identifier services, but also provide various search tools for their holdings. CLARIN as a whole provides a federated content search where you can search the holdings of different centers. Centers also provide training and dissemination and usually support a particular user community. CLARIN as a whole collaborates with so-called "discipline-specific working groups," of which there are ten right now including fields as diverse as psychology, linguistics, history, and political science.

One of the major contributions of CLARIN so far is WebLicht (Web-based Linguistic Chaining Tool), an environment for constructing and executing linguistic annotation pipelines. WebLicht is built upon Service Oriented Architecture (SOA) principles, which means that annotation tools are implemented as web services that are hosted on servers at CLARIN centers. WebLicht allows researchers to mix-and-match tools from different centers, which is made possible by use of a common data exchange format Text Corpus Format (TCF), and by metadata concerning each individual tool, most importantly input/output specifications. A shared data exchange format allows the output of one tool in the pipeline to be used as input to the next, and tool metadata is used to guide the user to build only

[15] http://clarin.eu/
[16] http://www.perseus.tufts.edu/

valid pipelines. Currently WebLicht provides approximately 112 tools, most of which perform standard NLP tasks such as tokenization, part of speech tagging, detection of named entities, and morphological analysis. There is an authentication and authorization interface, called Shibboleth, which manages access control, because some of the corpora and tools have legal restrictions. CLARIN also provides WebAnno [23], which allows both manual and automatic annotation of data.

5.2.6. Tuesday, Session 1: Brainstorming Capabilities

The first goal for the discussion of the second day was to develop lists of potential capabilities that could solve the high-priority problems identified on the first day. There was no restriction on the kinds of capabilities that could be proposed: they were encouraged to be easy or hard, pedantic or creative, incremental or visionary. The only question of concern at this stage was: Is this something that would solve the problem? Participants were encouraged to think expansively, without too much regard to our current ability to implement. The idea behind this brainstorming session was to get all the good ideas on the table; they could be pruned later if necessary by practical constraints. The only caution that I provided was that we should not be too specific in describing the capabilities, as this would overly constrain us going forward and potentially mire the discussion in disagreements over feasibility. For each high-priority problem, I have organized and condensed the discussion into an unordered list of potential capabilities.

Difficulty finding and evaluating appropriate tools and resources for the task at hand. Participants emphasized that the problem of evaluation was of equal or greater importance to the problem of discovery.

- **Registry or Catalog of Tools.** There have been many attempted catalogs of annotation tools, and there are a number of existing resources that include lists of tools; for example, DiRT at Berkeley[17], the LRT inventory[18] (in Europe; CLARIN/MetaShare), or the LRE Map[19]. But these fall short in a number of ways. Steve Cassidy pointed out the example of Orbit[20], a registry for biomedical tools, which is based on a content management solution. That, he suggested, may be a slightly better model than just websites with lists. By extension, and by analogy with previously established type registries such as that in UIMA, several participants noted that we need more **formal and precise** descriptions of the features and capabilities of the tools. It was also noted that the best registries have built-in **automatic completeness checks** for the metadata, and an established **quality assurance** (QA) process when curating the data.
- **Linguistic Travel Planner.** By analogy with popular travel planning websites like Orbitz or Kayak, it was suggested we need a "linguistic" travel planner to plan a route between a set of desired inputs and a set of desired outputs, preferably allowing specific waypoints. It was noted that this would require standard metadata for tools.
- **Freeform Comments**. For whatever list, catalog, or registry of linguistic annotation artifacts is implemented, Michael Kipp suggested that it was critical to have the ability for the community to leave freeform comments. By observation of popular e-commerce websites like Amazon.com, he noted that the best way to evaluate unknown products is through user reviews, and the same principle applied in the linguistic annotation space. The ability to capture unstructured information opens the possibility of eliciting a large amount of useful, otherwise inaccessible, information. It would be great, too, if developers could comment back, and this facility would encourage tool improvement and maintenance.
- **Standardized Vocabulary.** Finally, the capability to describe tools in a standard vocabulary could prove useful to understanding what tasks the tool applied to. This capability was conceived as different from and less stringent compared to the ability to formally describe tools and schemes. What is intended here is the development of a common conceptual model of the tasks and sub-tasks of linguistic annotation, such that those seeking to do linguistic annotation can communicate clearly. There has already been some progress on this, in the form of the MATTER conceptualization [24].

Lack of support for development, documentation, reuse, naming, and formalization of annotation schemes and guidelines.

- **Formal Type Registries & Vocabularies.** Participants vocally supported a capability that allowed formal definition of the "types" covered by particular annotation schemes. Reference was made to the now defunct

[17] http://dirtdirectory.org/
[18] https://www.clarin.eu/content/language-resource-inventory
[19] http://www.resourcebook.eu/
[20] http://maveric.org/orbit.php

ISOcat effort[21], which attempted to provide a uniform vocabulary of linguistic types that could be captured in annotation schemes. Also referenced was the ongoing LAPPS Grid effort and its associated web-service exchange vocabulary which allowed it to federate with language grids throughout the world. In general, the capability was described as an abstract type registry, with formal descriptions and a formal vocabulary for describing the semantics of types, which would allow the description, derivation, and implementation of mappings between types. In general, participants focused on the capability to formally encode the *semantics* of annotation schemes, which naturally relates to a definition of semantic categories underlying particular annotations.

- **Support for MAMA.** While this capability remained somewhat vague, it was noted that some sort of clear support for the so-called MAMA or the "babbling" phase of annotation design is a critical missing capability. The MAMA phase refers to Model-Annotate-Model-Annotate [24], where the annotation designers are iterating between developing an annotation model and applying that model to data, thereby converging on a model that can be reliably applied to the data. The problem in this case is managing all the different versions and their differences and updating annotations created under a prior version. Related to this was the ability to use an updated version of an annotation scheme to verify an annotated corpus. Related efforts that were mentioned in this context included NVivo[22], the Web Service Exchange Vocabulary [25], and RDA[23].
- **Registry of Type Conversions.** Finally, it was proposed that, before a formal type registry was created, progress could be made by providing a registry of available type converters. For example, various researchers have written stand-alone converters between one scheme and another, or particular tools are known to be able to import and export in two different schemes. Collecting these conversion facilities into a single catalog would begin to relieve some of the burden of moving between different annotations schemes, without going to the difficulty of describing exactly the formalities of the mappings.

Inadequate Importing, Exporting, Conversion.

- **Standalone Converter.** By analogy with type conversion, syntactic conversion is also a problem, and some participants wondered if it wasn't possible to create a standalone service (either as a downloadable program, or a web service) that could convert between different syntactic formats. Such a service would significantly extend the applicability of existing tools. The data exchange library SaltnPepper[24] was mentioned in this context, and well as the ANC2Go[25] facility, which converts between the GrAF format and many other formats.
- **Adapters from Specific Tools to Data Formats.** A quicker path than building an all-to-all converter would be to identify key tools where adapters could bring about great effect in allowing that specific tool to use a previously incompatible data or repository format.
- **Cluster Map of Tools.** Ever more simply, it was proposed to merely map out sets of tools that have (or should have) conversions between them: in effect, answering the question "what tools work together right now?"
- **Publication Quality Output**. Jeff Good mentioned that a capability that would be of use to many non-technical users of annotations would be the ability to output "publication-quality" formats, which could be directly inserted in manuscripts. Because of the difficulty and importance of formatting data for publication, he noted, many non-technical researchers choose highly inadequate tools (such as Microsoft Word) for the initial annotation that make this final step easier.

Lack of Workflow Support. Also related to the idea of supporting workflows was the more specific idea of supporting chaining or composing linguistic processing tools.

- **Standard Vocabulary for Workflows.** Several participants emphasized that managing the workflow of annotation projects was one of the most critical components to ensure efficient and effective production of data. And yet, we have no standard vocabulary for describing annotation workflows, either to find tools that apply to particular parts of one's workflow, or to communicate to other researchers how one accomplished

[21] http://www.isocat.org/
[22] http://www.qsrinternational.com/
[23] https://rd-alliance.org/
[24] http://corpus-tools.org/pepper/
[25] http://www.anc.org/software/anc2go/

one's own workflow. A standard vocabulary for describing workflows—such as the common stages, tasks, and capabilities of annotation projects—would be a simple yet significant capability.

- **Standalone Facility for Combining Annotation Layers.** With any normal annotation project involving multiple annotation tools, several participants suggested building a capability to integrate annotation layers generated from different tools. The facility would be tool agnostic and would allow stand-off annotations to be integrated together without regard to their source. In this context was mentioned WebLicht, CLARIN's web-based Tool for combining annotation layer [26]. Also mentioned was ANNIS-3, which has the notion of multiple layers in the final product [27], and MMAX, which has interesting visualization capabilities in this regard [28].
- **Linguistic Pipeline Interoperability.** A significant amount of discussion revolved around achieving interoperability between the many different linguistic pipeline systems that are currently available. This discussion was so extensive that I have broken it out into its own section (§5.2.7). For example, the LAPPS Grid which is NSF funded via Vassar, Brandeis, CMU, and the LDC: [15], uses JSON LD and RDF for inter-module communication and has federated or will federate with six other grids throughout the world (e.g., MetaShare). However, at the time of the workshop it did not yet communicate with other pipeline solutions like UIMA or CLARIN's WebLicht. I defer discussion of this capability to the next section.
- **Supporting Human-in-the-Loop.** While this facility remained underspecified, it was noted that while humans and machines could and should profitably work together in generating annotations, there is very little support for human-in-the-loop workflows. I pointed out that the Story Workbench [13], [14] was designed from the start to incorporate a tight feedback loop between automatic and manual annotation and could potentially be a model in this regard.
- **Generic Workflow Support.** Finally, many participants desired a generic ability to describe precisely and implement easily annotation workflow. Right now, any annotation project of even mild complexity involves a number of people (designers, managers, adjudicators, annotators, etc.) operating over several sets of tools. The ability to quickly set up—or port from a pre-existing project—a workflow design for such a collaborative manual annotation project across such human and technical resources would be a major boon. Several participants noted that any workflow management system must include some ability to manipulate multimedia and multimodal data if it is to have wide applicability.
- **Portable Workflow Definitions.** In the absence of a generic workflow solution, and in the presence of multiple individual workflow management suites, perhaps an intermediate step would be to provide way of porting workflows across different sites. It was suggested that the LAMUS[26] tools, part of Language Archive Technology Suite, would be applicable to this capability.

Lack of User, Project, and Data Management. For many the capabilities listed against this problem, several participants noted that both WebAnno [23] and WebAnn [29] already implement them to some degree. Both allow some modicum of UI design, allow assigning of annotation artifacts to users, and control access. WebAnno works primarily with text, and WebAnn is more agnostic with regard to medium.

- **Generic Facilities for Searching Annotations.** No matter what kind of annotation is being performed, usually the ability to search those annotations is a useful facility. Most annotation tools, however, do not provide this capability, and those coming to the data later are even more poorly supported.
- **Generic Facilities for Versioning Annotation Schemes.** As noted above in "Support for MAMA", annotation schemes and annotated data usually go through several versions before being released to the public. We need tools and facilities that ease the management and control of these versions. A related tool in this regard is COMA, which is part of ExMARaLDa [30], which provides some version management capabilities.
- **Link between UI and Formal Scheme Definitions.** Related to the idea of "tool-building tools," which was a medium-priority problem, was some way of having UI functions and capabilities derive automatically from a formal definition for a scheme. One participant made the point that there are only a small number of UI ways to annotate, for example, a Token, and if an annotation scheme annotates tokens perhaps a good start to a UI could be automatically generated from the scheme itself.
- **Bridge between Standard Annotation and Crowdsourced Annotation.** Finally, it was suggested that some capability to automatically bridge between or port over standard annotation tasks to crowdsourced annotation would be of great value across the linguistic annotation space. Current models of local

[26] https://tla.mpi.nl/tools/tla-tools/lamus/

annotation don't scale to large volumes of annotators or data; right now, typically when you want to crowd-source an annotation task, you must custom-make a lot of software. While there has been some discussion of best practices for crowd-sourced annotation [31], there is still no generalized ability to distribute standard annotation tasks easily to crowdsourcing platforms.

5.2.7. Tuesday, Session 2: Linguistic Pipeline Interoperability Workshop

During the discussion of capabilities, there was a significant sidebar on enabling interoperability between the various linguistic pipeline solutions. Some major solutions available include the LAPPS Grid, WebLicht, UIMA, and LingPipe[27], plus a host of more focused efforts such as the Stanford CoreNLP suite or OpenNLP[28]. All of these platforms do similar things, but don't work together.

The problem that was noted was that once a researcher commits to a particular pipeline platform, it is extremely difficult to integrate tools from the other pipeline solutions, and extremely difficult to move to a different platform later.

Several participants suggested a focused effort to enable interoperability, as a sub-effort or parallel effort to any result of the UAT Workshop. Several suggested that a pipeline interoperability effort would be best embodied in a week-long technical workshop, where implementers of the major pipelines came together to agree on interoperability standards and interchange protocols, and implement those solutions for their pipelines, such that components could be shared across pipelines and pipelines could speak with one another. This workshop would be a true "working" workshop, where the expectation is that people would actually write code and have working systems at the end of the meeting. Several participants suggested that finding funding to have such a workshop should be easy, and in any case many of the pipeline implementers might be willing to come under their own funds.

Three general ideas about organization were discussed in relation to this specific effort. First, is that perhaps such a linguistic pipeline interoperability effort would be well served by a steering committee that could meet remotely (via video chat) in advance of the workshop, to hammer out the main specifications and write a guiding document. Eric Nyberg gave the example of the UIMA steering committee, which never met in person, but had bi-weekly phone calls and all the members were committed to writing a document together.

The second idea was that perhaps such an effort should be conducted under the auspices of an existing umbrella organization, such as RDA[29], SIGANN[30], or OASIS[31] (which is where UIMA started). A standards body like ISO was considered to be a poor fit, given the closed nature of the standards produced. But Eric Nyberg did suggest that interacting with a body like OASIS that has experience in creating these standards can be extremely useful, in that they can provide a lot of advice about how to write open specifications that academics are normally completely ignorant of.

Finally, the idea of a roadmap was suggested for organizing implementation and progress on such an effort. Such a roadmap would need to be in place before any implementation workshop, to guide the work and assess how much progress was made.

In the end, it was agreed by most participants this was an excellent idea that should be pursued in parallel, and not necessarily as a sub-part of, the related UAT workshop goals.

5.2.8. Tuesday, Sessions 2 & 3: Management and Sustainability

After discussion of the capabilities needed for the infrastructure, we proceeded to consider how such an infrastructure would be funded, managed, and sustained, and what partnerships would be necessary or appropriate. First I noted that the workshop itself was funded under NSF infrastructure planning funding and that I had promised in the original proposal that we would pursue the full Community Research Infrastructure grant to actually build the initial implementation. The real question of this session, then, was how to sustain it over the long term: funding cycles are three years, but the useful life of this infrastructure, if done right, could be 10-20 years or more. How do we support and manage that across changing funding priorities and individual researcher interests?

[27] http://alias-i.com/lingpipe/
[28] https://opennlp.apache.org/
[29] https://rd-alliance.org/
[30] http://www.cs.vassar.edu/sigann/
[31] https://www.oasis-open.org/

Discussion at first kept returning to the problem of finding initial funding to build the infrastructure. Presumably because this is something all the researchers know well: how one pitches an idea and works the system to get a three-year project funded. Discussion at one point turned to how much initial funding is obtainable or reasonable. At this point I interjected and refocused the discussion on the key question:

> *"…it is not so much about the amount of funding. We don't need $5 million up front. It's more about the sustainability. If we had $100,000 for ten years that would almost be better than $5 million for two years. I'm actually not that concerned about finding funding for the next three years. I think someone who formulates a good proposal will be able to find someone to give them money. The question is how to keep it alive over ten years. [The program manager who gave you the money initially] is not going to be there in four years. We're talking about sustainability over ten years, where people change, supporters change, programs move away. I agree if you have a single champion to go to the mat and spend a lot of time looking for funding at every new funding cycle, then, yeah, you're probably going be able to keep it alive. But individual research priorities change. This is one of the problems that we were talking about yesterday, I get a tool that a graduate student built, and that student graduates, so now that tool is not supported and there's no one on the back end to ask about it, and it dies. How do we prevent this from happening with the infrastructure?"*

This refocused participant's thoughts, and it was noted that the first question to ask in this regard was: how "heavy" is the infrastructure? Is it lightweight, able to be managed by a few people? Or is it heavyweight, requiring a lot of centralized institutional support? Participants clearly preferred the second option. I noted that this is exactly where we need to begin for sustaining over the long term, and we also need to start thinking about how many people, how many servers, and what institutional support will be needed to be sustained for ten years. At this point, the discussion turned to consider a number of funding agencies in turn and how their strategies matched up with long-term sustainment funding.

NSF. It was agreed that the U.S. National Science Foundation would be able to supply initial implementation funds over at least a three-year period. But the worry, then, was that the money would run out. How feasible, then, is it to move the support of the infrastructure around within NSF, or within the CRI program itself? At this point the idea of "multiple pillars" began to take shape, of supporting the infrastructure with a number of different, complementary sources. This would require, additionally, a number of different partnerships. The idea was floated of seeking a second planning grant before going for the implementation phase of the CRI.

NEH. The U.S. National Endowment for the Humanities, and the Office of the Digital Humanities (ODH) was mentioned as a possible support. Although it was noted that even though the humanities is not quite in the core domain of the target, they still use significant language resources. It was concluded that, because the NEH and ODH have so few resources, that they would make a good partner but perhaps were not the ideal source of primary funds.

NIH. One participant thought that it was very difficult to get infrastructure funding from the U.S. National Institutes of Health. In the NIH model, you must apply to a specific institute, but the institutes are disease oriented, and you have to show how your project is needed specifically to solve a particular disease problem. Another participant, however, noted that the NIH has funded infrastructure projects in the past, bringing up examples of various bioinformatics databases, and the Databrary project[32]. It was admitted that the NIH does sometimes fund projects of this sort, especially through the National Library of Medicine (NLM) which is the only institute that is somewhat broad and funds informatics related projects. On the downside, NLM has only a very small amount of money and in the past year they funded only 13 grants. I noted that perhaps a cross-institute call like the Early Stage Development of Science and Technology R01[33] might be appropriate here, but again, the disease-specific nature of the funding was considered to be a major barrier.

DARPA. Stephanie Strassel, who has deep experience with DARPA funding, noted emphatically that DARPA is not going to directly fund an infrastructure development project. The only way to obtain DARPA money in support of an infrastructure effort is to make a case that an enhancement of the infrastructure is critical to the success of an existing DARPA program, probably with regard to data collection. The case would need to propose, for example, that developing a module for an *already well-established* infrastructure would allow us to perform better some task that DARPA is already funding. For example, the new module would allow us to do a particular kind of human or automatic annotation more efficiently, accurately, cost-effectively, in more languages, or make it more useful to do

[32] https://databrary.org/

[33] http://grants.nih.gov/grants/guide/pa-files/PA-14-155.html

research in general. DARPA has been occasionally willing to fund these kinds of infrastructure enhancements funding in the past, but they must be tightly linked to the program technology goals. Any benefits for the community are side effects. As an example of how you might leverage current DARPA interests, Eric Nyberg mentioned the LORELEI program[34]. LORELEI stands for Low-Resourced Languages for Emergent Incidents, and the Program Manager, who is a friend to computational linguistics, might be open to giving some partial seed funding, especially if matched with NSF.

IARPA. The U.S. Intelligence Advanced Research Projects Agency was almost immediately eliminated by several participants as a potential funder. The reasons for this elimination are unclear.

Cross-Atlantic Coordination. It was suggested perhaps, given both the strong American and European presence at the workshop, that there was a way of stimulating coordination between U.S. and European funding agencies for a joint program, with the interest of each agency being reinforced by the interest of the other. It was noted by Nancy Ide that some had tried in the late 90s to get the NSF and EU to come together for a joint trans-Atlantic program around language technologies, but the effort stalled. An easier path, it was suggested, was to pursue project-level funding simultaneously: the RDA, for example, has managed to get funding from both the European Commission and the United States.

Overall, the discussion of potential U.S. funding agencies resulted in participants noting that there existed a general model that has been effective to sustain similar efforts in the past. The model is that organizations like the NSF fund the core infrastructure, backend, and long-term maintenance, but then there are specific technology-driven programs (such as from DARPA or the NIH) that fund an extension to address a particular need.

The discussion then moved to non-Federal funding opportunities.

Existing Non-Profits. It was agreed that such an infrastructure would clearly need to partner with existing institutions like the Linguistic Data Consortium (LDC), SIL, and other non-profit organizations already in the business of providing infrastructure and tools. However, it was determined that these organizations would not be in a position to provide any significant sources of funding, except perhaps exchanges in kind.

Corporate Collaboration. Taking inspiration from the UIMA model, I suggested that perhaps there was an opportunity for corporate funding from a place like IBM, Google, or Microsoft. Google provided significant funding for a long time for Mozilla[35], and IBM developed and continues to support the Eclipse project[36]. For IBM, Eric Nyberg noted that the argument for funding would be similar to that to DARPA: the infrastructure would have to be critical for something they were already doing, and they probably would not fund the initial development. Several participants thought that Google probably would not be interested, and that they probably would be primarily interested in building it themselves and owning it for internal use, rather than funding any community development. Mention was made of Google.org, the philanthropic arm, and prior efforts to make endangered languages more accessible: however, it was noted that they were difficult to work with and generally didn't play well with others.

Non-Government Sources. Non-government funders would be one potential source, participants thought, but there were few specific proposals as whom to approach. The Mellon Foundation was mentioned, but it was noted that if you were not part of the prestige research university club then they were not generally interested in interacting with you. That meant that partnership with a prestige research university (e.g., CMU for the Mellon foundation), was critical to securing this sort of funding.

An idea that was mentioned during the closing dinner is worth mentioning here. It was suggested that, in order to keep inventories of tools up-to-date, an alternative to long-term funding is to make sure tool providers see value in keeping their artifacts well represented. This is the idea behind ISLRN and, even closer to this case, ResearchGate. One way to encourage this is to push the community to value a resource description in a catalog as another kind of publication. The idea here is that academics are forced to invest in what provides academic reputation, and sustainable software engineering does not (yet) fall into this category, and so increasing the value of these types of products might make sustainability an easier task.

The discussion now moved to considering different potential models for organization of the infrastructure, focusing on previously successful examples.

[34] http://www.darpa.mil/program/low-resource-languages-for-emergent-incidents
[35] https://www.mozilla.org/
[36] http://www.eclipse.org/

Galaxy. A project that had come up several times already was the Galaxy project, which is a workflow engine in bioinformatics. After initial funding to establish the infrastructure, it has since grown to be indispensable for that community, and the infrastructure has become somewhat self-sustaining: there are several institutions that have decided that it is critical to their work, and they are pitching in a small amount of money. Steve Cassidy noted that even the Australian funding body is pushing money into Galaxy.

CMU QA Consortium. Eric Nyberg mentioned his CMU QA consortium as a potential model, albeit one that is closely tied to a particular researcher and institution. In this model, all the services provided by the consortium are free for everybody. But sponsors provide funds in exchange for the CMU team doing some work to build out part of the infrastructure to their requirements, or teach them how to use it. These funds are usually sufficient to cover a graduate student and have included institutions like Rosch, IBM, Boeing, and the Government of Singapore. He noted that this was basically the RedHat model[37]. The model, however, is tricky: you must find the right sponsors who like open source software and have a real need you can fill. If you can do this, you can even fund sometimes permanent people like staff programmers. But the funding is term-by-term: they could all decide not to renew, and then four or five students suddenly lack funding. It's risky, but it works for this particular project.

Linguistic Data Consortium. The LDC was offered by several participants, including Stephanie Strassel, as a potential model. LDC has a hybrid sustainment model: there is a very small core group that has been sustained over twenty-five years, accompanied by a scalable staff that grows or shrinks in response to project funding. At times the LDC has been able to sustain two or three developers; at other times, there are none. This goes back to the model mentioned above where there's a low-level sustained, long-term funding for the core infrastructure, and then opportunistic funding for extensions to the infrastructure, tool modules or other kinds of extensions, new import-export capabilities, based on particular project funding or corporate funding. It was also noted that LDC would make a perfect partner, especially as it has already demonstrated its longevity (which is a good predictor of future success).

Subscription. I floated the idea of a subscription model, like many open source projects, which solicit small, one-off payments from users. For example, the project ask every user for a voluntary, non-obligatory, contribution of, say, $30. In general, however, the participants were opposed to this, and they noted that the current trend is toward completely open and free tools.

LINGUIST List. Finally, Jeff Good mentioned the LINGUIST list[38] as a potential model. Although primarily an information distribution organization, they have a foundation, a board of directors, and a fund drive. Their funding comes partly from universities, partly from the corporate fund drive, and partly from individual donations, but it shows how with application and general appeal a small funding base can be sustained over the long term.

The discussion of management and sustainability concluded with three final ideas. **First** was an interesting idea from Jeff Good, who noted that the Language Sciences Press[39] made the problem of sustainability a research question that was to be deferred to the next stage. To do this, they brought an economist on staff to look at the long-term sustainability of the open access publishing. This was thought to be potentially a viable option, because it would be easy to get a substantial pot of money for the first three years

Second, one participant wondered whether the word "platform"—rather than infrastructure—was perhaps a better way to describe the solution we are conceiving: a set of interrelated and mutually supportive pieces that come together to allow others to surmount the problems of balkanization. This idea was well-received but not discussed in depth.

Third, we discussed the question of whether or not the infrastructure should have its own "umbrella" organization that would host it, among other things, or if it should become an organizational sub-part of an existing organization. The general feeling of the participants was that creating yet another umbrella organization was not needed or advised, and would just complicate matters. Further, if the infrastructure was explicitly tied to an existing country- or region-specific project (e.g., CLARIN) it might take on the flavor of the region, discouraging global use. While participants were agnostic about subsumption under existing global umbrella organizations (e.g., RDA), the general feeling was that we shouldn't create bureaucracy or administration where it is not absolutely needed, and the question of the umbrella could be safely deferred until something had been built.

[37] https://www.redhat.com/
[38] http://linguistlist.org/
[39] http://langsci-press.org/

5.2.9. Tuesday, Sessions 3 & 4: Final Recommendations

The segment of the workshop was devoted to crystallizing the whole discussion of the previous two days into a set of clear, actionable recommendations. The idea of these statements is that they could be seen as the explicit recommendations of an international group of experts on what to accomplish next.

Before brainstorming on specific statements, I put on the screen a graphic provided by Nancy Ide and James Pustejovsky (reproduced in Figure 2), from their recent LAPPS-Grid-related NSF CRI proposal. I noted that that each of the uncolored boxes in the figure represents a major capability we have been discussing, and further shows how they all might plausibly work together.

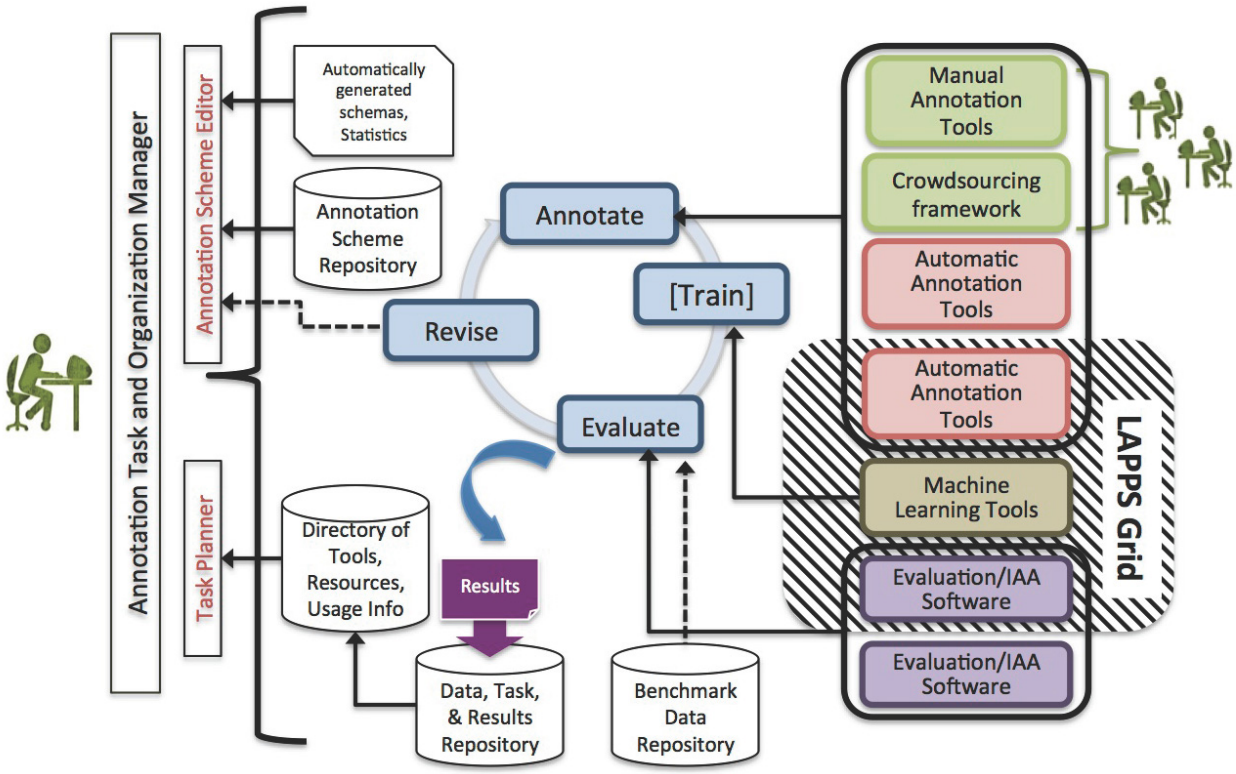

Figure 2: Component diagram of the Linguistic Annotation Service Architecture, proposed by Ide & Pustejovsky to NSF in 2014.

After consideration of this diagram, the discussion moved to brainstorming recommendations, of which the group produced 16. For each of the recommendations below, the bold-faced text was typed on the screen during the workshop and the wording debated and agreed to (or at least, not objected to) by a majority of the participants. The list is organized by general topic, and not by priority: no priority was explicitly attached to the recommendations, but a priority can be roughly inferred by matching recommendations against problem priorities. I created the topics as an organizing framework while writing this report.

Planning

There were six recommendations revolving around planning and information gathering activities.

1. **Conduct a survey of practitioners as to what they are actually using: how many are using tool X, how many people are not using any tool at all?** Recipients of the surveys could be identified by extracting emails from research papers involving linguistic annotation.
2. **Identify gaps in the current landscape of tools: what tools are missing, what features are missing from existing tools?** This information could perhaps be collected via questionnaires to both practitioners and tool-builders, as in the previous recommendation.

3. **Develop clear use cases that illustrate our target stakeholders, demonstrate what will be gained by working on these solutions, and consider combinations of domains, media types, schemes**. The use cases should ideally also involve broadly scoped, shared task evaluations.
4. **Consider which emerging best practices for data rights management are most applicable.** Data rights management are critically important when sharing annotated data, of which there may be quite a bit if solutions involve cataloging annotation schemes and tools and their link to the actual data sets. There are also a number of emerging data rights management practices (e.g., Creative Commons[40], among many others). Which emerging best practices for data rights management are most applicable to linguistic annotation?
5. **Develop a roadmap.** Develop an explicit plan (beyond this technical report) for moving forward to solving the balkanization problem, as to what the tasks are, what their interrelationships are, and by when they should be accomplished. Developing a roadmap is an involved process, and may need the attention of a working group over some period of time.
6. **Hold a standard workshop series.** There are researchers who are working in solving problems in this area, as well as those who have insights and other recommendations. This suggests that perhaps an academic workshop with submitted or invited papers would be appropriate. The hosting venue would be critical, because researchers are unlikely to travel to just go a workshop. The Language Resources and Evaluation Conference (LREC) would be highly appropriate in this regard. One possible existing venue at which this topic could be discussed would be the OIAF4HLT workshop[41].

Describing & Cataloging

There were four recommendations revolving around describing and cataloging annotation resources. While "describing" is similar to the information gathering recommendations proposed in the previous category, the set of recommendations below was such a strongly distinctive set, which involved moving well beyond collecting information via surveys, that it seemed to warrant its own category.

7. **Assemble a catalog of the annotation tools, with metadata and features, coupled with a tool wiki; send out questionnaires for new resources/tools to add them to the catalog.** The key features of this recommendation involve a catalog listing annotation tools. It should have extensive, detailed metadata about tools, in a format tailored to annotation tools. It should also involve a wiki, or some other user-editable review system, that allows free text feedback and evaluation of tools. Another feature that was thought to be important is the ability to compare tools side-by-side with respect to a selection of features or capabilities: thus some sort of table visualization which allowed you to see tools on the left side and features/capabilities along the top. One problem that we discussed was that of making sure that the catalog is complete and stays up-to-date. To this end, it was suggested that a system for sending out questionnaires for new resources/tools found in the literature could help. A number of existing resources could be mined to provide an initial catalog population, including the LREC language resource roadmap[42] and ELRA data[43].
8. **Provide a catalog of the annotation schemes, with metadata coupled with a wiki; send out questionnaires for new schemes to add them to the catalog.** Similar to a catalog of annotation tools, the participants recommended that a catalog of annotation schemes be assembled, with similar functionality to the annotation tool catalog: free-text wiki or review capabilities, comparison capabilities, along with links to datasets and tools that support.
9. **Define an inter-lingua for terms like "annotation", "annotator", "artifact", "tool", "application", "workflow", "pipeline", etc. Give examples.** This recommendation covered several levels of application. In the most general application, participants recommended coming up with common definitions of common linguistic annotation terms, such as listed in the boldface above. Having a "dictionary" of these terms, of sorts, would enable researchers across disciplines who use linguistic annotation to speak clearly to each other about their needs. At a more specific level, there was some discussion of the need to be able to describe experimental setups in a way that was understandable (to encourage validation and reproducibility). In its more precise interpretation, this recommendation asks for a language that allows one

[40] https://creativecommons.org/
[41] http://glicom.upf.edu/OIAF4HLT/
[42] http://www.resourcebook.eu/
[43] http://catalog.elra.info/

to port workflows between tools and annotation projects: a way of writing down precisely the workflow of a whole annotation project, including the tool configuration, user assignments, text assignments and so forth, so that the exact same workflow can be applied again in a reproducible way.

10. **Develop a formal language for writing down the operational or mappable semantics of annotation schema (contrast with UIMA type system: inclusion with semantics and ability to automatically reason; this should include difference between versions).** This recommendation embodies the desire for a way of describing, in formal, precise, and computer-checkable ways, the semantics of annotation schemes from both an operational and mapping standpoint. *Operational* here means "what the scheme means in practice when applied to text"; *Mapping* means "how the objects described by different schemes equate to one another". There was a clear acknowledgement that this would require not only some encoding of the annotation guide, description, or protocol used by human annotators, but also the logical level description of a scheme (e.g., this annotation can be any of five types, and they must have these X attributes to be well-formed). A start to such a language would allow a mapping into a common annotation type system like UIMA or GATE, or how some scheme can be mapped into another. Then it could be the mapping of that into UIMA, or GATE. Several participants believed that such a language should probably not be a once-and-for-all ontology of linguistic types, such as ISOcat or the Ontologies of Linguistic Annotation (OLiA)[44], and that building and maintaining such an ontology in the long term was probably infeasible. The participants then proposed alternative implementation models, including using concrete examples of annotations on language coupled with mappings between specific type systems, with either formal or informal descriptions of information loss that accompanies that transformation. Importantly, having a formally implemented type mapping system allows the system to be checkable: if you have a mapping F that maps type system A to type system B, then you can get an expert in B to check the mapping of F(A). Another comment that was added was that such a system would naturally allow you to manage versioning of annotation schemes, which is a problem: if you have type system N, but new information forces you to develop type system N+1, such a mapping facility would allow you to document the relationship: this was described as formalizing the annotation design MAMA or "babbling" process.

Implementation

Five recommendations focused on various aspects of implementing solutions.

11. **Hold a narrowly focused technical working group, ideally starting with a weeklong meeting, driven by a roadmap, to solve the problem of data interchange between annotation pipelines like WebLicht, the LAPPS Grid, Galaxy, UIMA, V3NLP, Gate.** Currently there are numerous linguistic processing pipeline solutions, and they all work with different or overlapping (but not identical) sets of tools. Switching to another pipeline is not a simple task, and when you choose one pipeline solution, you do not have access to tools that use the other pipelines. Participants highly recommended trying to find a way to bridge between different pipelines so all the underlying tools could be used everywhere. They emphasized that this was not necessarily a file format, but more generally some method of moving data between the pipelines. Participants focused on having this effort run in parallel with other recommendations and cast it as a straightforward way to get something done quickly and immediately. If such cross-pipeline bridges are eventually subsumed under some sort of formal type-mapping system, this was considered a great bonus, but the development of the type-mapping system should not be required before constructing pipeline interconnects.
12. **Develop best practice guidelines for overall annotation project manage, calculation of IAA, creation of gold standard data, etc.** Several participants noted during the workshop that there are numerous pockets of expertise on best practices in annotation, but they are not accessible to many researchers. Topics that could be treated by best practice guides included: annotation scheme development; annotation project management; training of human annotators; calculation of inter-annotator agreement; methods for automatic and manual annotation; creation of gold standard data; evaluation metrics for both automatic manual annotation; and, more generally, resources and methods.
13. **Ensure that tools and formats are "stable" with respect to uninterpretable information.** Moving data between multiple tools and pipelines raises the question of what those systems should do when they encounter information that they cannot use or interpret. This recommendation speaks to establishing a standard expectation or best practice that systems do not modify information that they cannot interpret: by

[44] http://nachhalt.sfb632.uni-potsdam.de/owl/

analogy with a stable sort algorithm, if the system does not have reason to touch a part of data, it should not, and maintain the integrity of that part.

14. **Explore the possibility of defining an abstract API for extracting, referencing, depositing annotations from/in data and annotation stores and archives that covers existing APIs like Alveo or SRU (should have a web based interface).** Unifying ways of accessing data, regardless of its physical location, was seen a particularly valuable potential implementation. An example use case is: Researcher A has a license to a corpus from LDC. An access API would allow him to download the corpus programmatically, rather than by hand. Further, if he creates a set of new annotations over that corpus, the API would allow him to store those new annotations as linked to the original corpus data, but not actually on an LDC server with the original files. These cross-indexed annotations would then be available anywhere in the world through the API, to those with the proper access rights.
15. **Ensure that any solution accommodates humans-in-the-loop (including crowdsourcing) in a flexible, integrated way.** This recommendation speaks to establishing best practices that properly incorporate humans-in-the-loop. Integrating humans and machines together in annotation workflows is currently awkward: current tools do not manage this transition well.

Other

16. **Encourage funding agencies to find ways to generate trans-oceanic partnerships between themselves to fund coordinated activities.** The goal here would be to leverage the international nature of such an infrastructure, and find a way for funding agencies across continents to support each other. While some participants were skeptical this could ever even get started, Nancy Ide noted that this actually did happen in 1998, where U.S. and EU program managers did get together and create a framework to fund these trans-Atlantic projects; unfortunately, the framework hit some snags and never got off the ground.

6. Conclusions

The problem of the balkanization of annotation tooling is major barrier to human language technologies and natural language processing today. The inability of researchers to quickly and easily identify, use, and combine the tools they need to accomplish linguistic annotation results in blocking of research progress, lost opportunities for new discoveries, delayed or diminished results, and significant waste of resources.

The workshop produced quite a bit of excellent discussion centered around these problems, and this technical report has laid out the step-by-step progress of that discussion. The workshop participants identified ten major problem classes, of which five were deemed high-priority. Across the five high-priority problems, participants identified twenty-one desired capabilities that would help mitigate or eliminate that problem. Finally, participants proposed sixteen detailed recommendations (shown in §5.2.9) for next steps.

It is clear that there is much work to be done. The workshop generated a large number of very good ideas, and it will require many years and international cooperation to effect. The first step (already underway) is to seek funding to start an effort and begin to implement these recommendations. Hopefully, in years to come, we will see the vision of this workshop come to fruition and make real progress in solving—or possibly even eliminating—the balkanization of annotation tooling.

7. Acknowledgements

The work described in this report was funded under NSF Community Infrastructure Planning (CI-P) Grant #1405863 to MIT, and #1536043 to FIU. Thanks to Sharon Mozgai, my research assistant who helped assemble the bibliography of annotation tools for the first task of the grant. Thanks to the many unnamed researchers who provided excellent feedback during the workshop planning stage. Finally, thanks to my students Victor Yarlott and Joshua Eisenberg for proof reading and feedback on earlier drafts of this report.

8. References

[1] S. Bird and M. Liberman, "A Formal Framework for Linguistic Annotation," *Speech Commun.*, vol. 33, no. 1, pp. 23–60, 2000.

[2] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The Penn Treebank: Annotating Predicate Argument Structure," *ARPA Hum. Lang. Technol. Work.*, 1994.

[3] M. Palmer, P. Kingsbury, and D. Gildea, "The Proposition Bank: An annotated corpus of semantic roles," *Comput. Linguist.*, 2005.

[4] J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, and G. Katz, "TimeML: Robust Specification of Event and Temporal Expressions in Text," in *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, 2003, vol. 5.

[5] R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky, "TimeML Annotation Guidelines, Version 1.2.1," 2006.

[6] H. Cunningham, D. Maynard, and K. Bontcheva, *Text Processing with GATE (Version 6)*. London, UK: University of Sheffield, 2011.

[7] G. Petasis and V. Karkaletsis, "Ellogon: A new text engineering platform," in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, 2002, pp. 72–78.

[8] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "brat: a Web-based Tool for NLP-Assisted Text Annotation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012): Demonstrations*, 2012, pp. 102–107.

[9] S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: A Unified Relational Semantic Representation," in *Proceedings of the International Conference on Semantic Computing*, 2007, pp. 517–526.

[10] W. N. Francis and H. Kucera, "Brown Corpus Manual," Providence, RI, 1979.

[11] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell, "GATE Teamware: a web-based, collaborative text annotation framework," *Lang. Resour. Eval.*, vol. 47, no. 4, pp. 1007–1029, 2013.

[12] K. Bontcheva, I. Roberts, L. Derczynski, and D. Rout, "The GATE Crowdsourcing Plugin: Crowdsourcing Annotated Corpora Made Easy," in *Proceedings of Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014, pp. 97–100.

[13] M. A. Finlayson, "Collecting Semantics in the Wild: The Story Workbench," in *Proceedings of the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence (published as Technical Report FS-08-06, Papers from the AAAI Fall Symposium)*, 2008, vol. 1, pp. 46–53.

[14] M. A. Finlayson, "The Story Workbench: An Extensible Semi-Automatic Text Annotation Tool," in *Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4)*, 2011, pp. 21–24.

[15] N. Ide, J. Pustejovsky, C. Cieri, E. Nyberg, D. Wang, K. Suderman, M. Verhagen, and J. Wright, "The Language Application Grid," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[16] N. Ide and K. Suderman, "GrAF: A Graph-based Format for Linguistic Annotations," in *Proceedings of the Linguistic Annotation Workshop*, 2007, pp. 1–8.

[17] N. Ide and L. Romary, "International standard for a linguistic annotation framework," *Nat. Lang. Eng.*, vol. 10, no. 3–4, pp. 211–225, 2004.

[18] M. Dickinson, "Detection of Annotation Errors in Corpora," *Lang. Linguist. Compass*, vol. 9, no. 3, pp. 119–138, 2015.

[19] K. Krippendorff, *Content Analysis: An Introduction to its Methodology*. 2004.

[20] E. Afgan, D. Baker, M. van den Beek, D. Blankenberg, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, C. Eberhard, B. Grüning, A. Guerler, J. Hillman-Jackson, G. Von Kuster, E. Rasche, N. Soranzo, N. Turaga, J. Taylor, A. Nekrutenko, and J. Goecks, "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update," *Nucleic Acids Res.* , May 2016.

[21] S. Barbara, "EUDICO Linguistic Annotator (ELAN)," *Lang. Doc. Conserv.*, vol. 1, no. 2, pp. 283–289, 2007.

[22] F. Schiel, "MAUS Goes Iterative," in *Proceeindgs of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, 2004, pp. 1015–1018.

[23] Y. Seid Muhie, I. Gurevych, R. E. de Castilho, and C. Biemann, "WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013): System Demonstrations*, 2013, pp. 1–6.

[24] J. Pustejovsky and A. Stubbs, *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. Sebastopol, CA: O'Reilly, 2013.

[25] N. Ide, J. Pustejovsky, K. Suderman, and M. Verhagen, "The Language Application Grid Web Service Exchange Vocabulary," in *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT)*, 2014.

[26] E. W. Hinrichs, M. Hinrichs, and T. Zastrow, "WebLicht: Web-Based LRT Services for German," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010): System Demonstrations*, 2010, pp. 25–29.

[27] A. Zeldes, J. Ritz, A. Lüdeling, and C. Chiarcos, "ANNIS : a search tool for multi-layer annotated corpora," in *Proceedings of Corpus Linguistics 2009*, 2009.

[28] C. Müller and M. Strube, "Multi-level annotation of linguistic data with MMAX2," in *Corpus technology and language pedagogy: New resources, new tools, new methods 3*, 2006, pp. 197–214.

[29] C. Cieri, D. Dipersio, M. Liberman, A. Mazzucchi, S. Strassel, and J. Wright, "New Directions for Language Resource Development and Distribution," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[30] C. Meißner and A. Slavcheva, "EXMARaLDA review," *Lang. Doc. Conserv.*, vol. 7, pp. 31–40, 2013.

[31] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl, "Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014, pp. 859–866.

Appendices

A. Workshop Participants & Demographics

The workshop was attended by 23 researchers from around the globe.

#	Name	Affiliation	Country	Website
1.	Dr. Claire Bonial	UC Boulder	U.S.	https://www.researchgate.net/profile/Claire_Bonial
2.	Prof. Steve Cassidy	Macquarie U.	Australia	http://comp.mq.edu.au/~cassidy
3.	Prof. Wendy Chapman	U. Utah	U.S.	http://medicine.utah.edu/faculty/mddetail.php?facultyID=u0073209
4.	Prof. Markus Dickinson	Indiana U.	U.S.	http://cl.indiana.edu/~md7
5.	Prof. Mark Finlayson*	Florida Int. U.	U.S.	http://cs.fiu.edu/~markaf
6.	Prof. Jeff Good	U. Buffalo	U.S.	http://www.acsu.buffalo.edu/~jcgood
7.	Dr. Thomas Hanke	U. Hamburg	Germany	http://www.sign-lang.uni-hamburg.de/dgs-korpus/index.php/project-staff.html
8.	Prof. Erhard Hinrichs	U. Tubingen	Germany	http://www.sfs.uni-tuebingen.de/~eh
9.	Marie Hinrichs	U. Tubingen	Germany	http://www.sfs.uni-tuebingen.de/ascl/mitarbeiter/mitarbeiter-detail/hinrichs-marie.html
10.	Prof. Nancy Ide	Vassar College	U.S.	http://www.cs.vassar.edu/~ide
11.	Prof. Michael Kipp	Augs. U. Appl. Sci.	Germany	http://www.michaelkipp.de
12.	Prof. Brian MacWhinney	CMU	U.S.	http://psyling.psy.cmu.edu
13.	Dr. Diana Maynard	Sheffield	U.K.	http://www.dcs.shef.ac.uk/~diana
14.	Prof. Eric Nyberg	CMU	U.S.	http://www.cs.cmu.edu/~ehn
15.	Dr. Georgios Petasis	NCSR Demokritos	Greece	http://www.ellogon.org/petasis
16.	Prof. James Pustejovsky	Brandeis	U.S.	http://jamespusto.com
17.	Prof. Anna Rumshisky	U. Mass. Lowell	U.S.	http://www.cs.uml.edu/~arum
18.	Dr. Gary Simons	SIL International	U.S.	http://www.sil.org/~simonsg
19.	Han Sloetjes	MPI Nijmegen	The Netherlands	http://www.mpi.nl/people/sloetjes-han
20.	Dr. Brett South	U. Utah	U.S.	http://medicine.utah.edu/bmi/people/staff-technical/index.php
21.	Dr. Pontus Stenetorp	U Tokyo	Japan	http://pontus.stenetorp.se
22.	Stephanie Strassel	LDC	U.S.	https://www.ldc.upenn.edu/staff/stephanie-strassel
23.	Dr. Marc Verhagen	Brandeis	U.S.	http://www.cs.brandeis.edu/~marc

Table 2: List of workshop participants. *=workshop organizer

Approximately 3/5 of the workshop attendees worked at institutions in the United States.

Region	Number	Fraction
U.S.	14	61%
Non-U.S.	9	39%
Europe	7	30%
Asia & Pacific	2	9%
Total	23	

Table 3: Breakdown of participants by region of institutional affiliation

Approximately 90% of the workshop attendees held the Ph.D. degree. Approximately 50% were professors.

Position	Number	Fraction
Professor	12	52%
Doctoral-Level Researcher	8	35%
Non-doctoral-level Researcher	3	13%
Total	23	

Table 4: Breakdown of participants by position.

B. Participant Biographical Sketches

Mark Finlayson (Organizer)

Assistant Professor
School of Computing and Information Sciences
Florida International University, United States
markaf@fiu.edu http://cs.fiu.edu/~markaf

Professor Mark Finlayson received his Ph.D. in 2012 from MIT in Artificial Intelligence and Cognitive Science. His research focuses on the science of narrative, including understanding the relationship between narrative, cognition, and culture, developing new methods and techniques for investigating questions related to language and narrative, and endowing machines with the ability to understand and use narratives for a variety of applications. He has worked on linguistic annotation in service of this research, building the **Story Workbench**, an annotation tool designed to allow the simultaneous manual, automatic, or semi-automatic annotation of over 20 different layers of syntax and semantics onto text by non-technical annotators. Using the Story Workbench he has collected and annotated a number of corpora of narratives, including the **UMIREC corpus** (UCM/MIT Indications, Referring Expressions, and Co-Reference), the **N2 Corpus**, and a deeply annotated corpus of Russian folktales. With his students he has built variety of widely used NLP tools in Java, including **JWI**, **jMWE**, **jVerbnet**, and **jSemcor**.

Claire Bonial

Research Associate
Department of Linguistics
University of Colorado at Boulder, United States
cbonial@me.com https://www.researchgate.net/profile/Claire_Bonial

Dr. Claire Bonial completed her Ph.D. in Linguistics and Cognitive Science at the University of Colorado, Boulder in 2014. Since 2007 Dr. Bonial has worked with Dr. Martha Palmer as a Research Associate in CU's Center for Language Education and Research (CLEAR) Lab. During this time Dr. Bonial has focused on the development, maintenance, and expansion of **PropBank**, **VerbNet**, **SemLink**, and the **Abstract Meaning Representation (AMR)** project. Dr. Bonial also assisted in the development of the **Jubilee** annotation tool, used for PropBank annotation, as well as **Cornerstone**, the tool used for the creating and editing the PropBank lexicon of frame files (i.e. sense inventory).

Steve Cassidy

Associate Professor
Department of Computing
Macquarie University, Australia
steve.cassidy@mq.edu.au http://web.science.mq.edu.au/~cassidy

Professor Steve Cassidy is a computer scientist (with a Ph.D. .in Cognitive Science) who has worked on various areas relating to speech and language technology over the last 30 years. With Jonathan Harrington he developed the **Emu** Speech Database System to support corpus based research in speech and acoustic phonetics. Emu supports a flexible hierarchical annotation system and provides a query language and analysis environment based on the R Statistical environment. Emu is widely used to support research on small and large scale speech corpora and includes tools to support every stage of the corpus collection and analysis lifecycle. Emu is now maintained by a team of developers in Munich. Professor Cassidy was also recently involved in the development and collection of an audio visual **Corpus of Australian English** from around 1000 speakers around Australia. He built the software for data capture and a server based system for data upload and publishing. His most recent work has been on the **Alveo Virtual Laboratory** which is both a repository for language resources and a platform to support tools for exploration and analysis of language data. Alveo currently holds around 20 collections including audio, video and text resources and is working on new acquisitions of data and tools.

Wendy Chapman

Chair & Professor, Department of Biomedical Informatics
University of Utah, United States
wendy.chapman@utah.edu http://medicine.utah.edu/faculty/mddetail.php?facultyID=u0073209

Professor Wendy Chapman has a B.S. in Linguistics, and earned her Ph.D. in Medical Informatics from the University of Utah in 2000. From 2000-2010 she was a NLM postdoctoral fellow and then faculty at the University of Pittsburgh, after which she joined the Division of Biomedical Informatics at the UC San Diego in 2010. In 2013, she became the chair of the University of Utah, Department of Biomedical Informatics. Professor Chapman's research focuses on developing and disseminating resources for modeling and understanding information described in **narrative clinical reports**. She is interested not only in better algorithms for extracting information from clinical text NLP but also in generating resources for improving the NLP development process (such as shareable annotations and open source toolkits) and in developing user applications to help non-NLP experts apply NLP in informatics-based tasks like clinical research and decision support. She has led development of several openly available clinical corpora and annotated corpora, including the **Pitt NLP Corpus** (unannotated clinical reports from 13 hospitals available for NLP research), the **ShARe Corpus** (clinical reports annotated with a multi-layer syntactic and semantic schema used in the CLEF/ShARe Shared Task 2013-2014 and SemEval Challenge 2015). She has helped develop a variety of annotation schemas that have culminated in the **Schema Ontology** and **Modifier Ontology** for annotating clinical text. She has also been involved in development of several annotation tools, including **e-HOST** for entities, attributes, and relations in clinical text and **Chart Review** for annotating complete patient records. She helped design a system for performing distributed annotation over private corpora (i.e., clinical reports) called **Annotation Admin**. Also, she has helped develop tools for visualizing NLP output, comparing automated annotations with manual annotations, drilling into errors using a visual interface, and providing user feedback to the NLP system based on identifying errors.

Markus Dickinson

Associate Professor
Department of Linguistics
Indiana University, United States
md7@indiana.edu http://cl.indiana.edu/~md7

Professor Markus Dickinson has worked extensively on two areas: 1) developing techniques to automatically detect and correct errors in different kinds of linguistic annotation; and 2) linguistically annotating corpora containing second language learner data. The former project, **DECCA** (Detection of Errors and Correction in Corpus Annotation), was an NSF-funded project and has led to a recent orthogonal project to **DAPS** (Detect Anomalous Parse Structures), with the goal of being able to build very large annotated corpora. The latter work is best exemplified by the **SALLE** (Syntactically Annotating Learner Language of English) project, an ongoing effort adding multiple layers of linguistic annotation.

Jeff Good

Associate Professor
Department of Linguistics
State University of New York at Buffalo, United States
jcgood@buffalo.edu http://buffalo.edu/~jcgood

Professor Jeff Good's current research interests include the linguistic typology of linear relations, the comparative morphosyntax of Niger-Congo, the documentation and description of Bantoid languages of the Lower Fungom region of Northwest Cameroon, and the role of emerging digital methods in the documentation of endangered and other low-resource languages. In this last area, he has been especially interested in the development of standards for encoding and annotating lexical and grammatical data and in tools to facilitate linguistic fieldwork. He has served as co-PI on the Lexicon Enhancement via the **GOLD Ontology project**, where he directed the conversion of thousands of wordlists stored in a legacy format into an XML format designed for interoperability, and as PI of the pilot **Pangloss project**, which explored the possibility of building an annotation tool within a word processing system. He

has additionally served as PI on a number of projects involving the documentation of **endangered languages** of Cameroon, and his current work in this area, also funded by NSF, has a collaborative component with a specialist in databases to build tools to support the management of data collected in the field. In 2014, he organized the **ComputEL workshop** to explore how computational linguists and endangered language linguists could more effectively collaborate http://buffalo.edu/~jcgood/ComputEL.html). In his work in linguistic typology, he has explored how graph-based descriptions of linguistic constructions can be rigorously compared with each other and developed prototype tools to support this. Within the linguistics community, he often serves as an informal liaison between the **language documentation** community and those developing digital standards for language resources.

Thomas Hanke

Researcher
Institute for German Sign Language and Communication of the Deaf
University of Hamburg, Germany
thomas.hanke@sign-lang.uni-hamburg.de http://dgs-korpus.de

Dr. Thomas Hanke works on language resources for **sign languages** and corresponding tools. He developed **syncWRITER** (1990), a first approach to annotate digital video. He also worked on **iLex**, a full-fledged team annotation environment for sign languages integration image processing. He is currently managing a long-term research project working towards a reasonably-sized **corpus of German Sign Language**, funded by the German Academies of Sciences program.

Erhard Hinrichs

Professor
General and Computational Linguistics
Tübingen University, Germany
erhard.hinrichs@uni-tuebingen.de http://www.sfs.uni-tuebingen.de/~eh

Professor Erhard Hinrichs is director of the Computational Linguistics research group at the University of Tübingen, Germany. He obtained a Ph.D. in Linguistics from The Ohio State University. His previous positions include Research Fellow at the Beckman Institute for Advanced Science and Technology; Assistant Professor at UIUC; and Research Scientist at BBN. His research interests include the computational modeling of language comprehension (particularly of syntax and semantics) and of language variation with special emphasis on the use of machine learning approaches to dialectology. Dr. Hinrichs has extensive experience in project coordination and leadership, e.g., as scientific coordinator of the **D-SPIN** project, member of the executive board of the **ESFRI** project **CLARIN**, and co-director and member of the **CLARIN-ERIC** Board of Directors. He has done key work on the annotation web-based linguistic annotation tool **WebLicht Tübingen Treebanks (TüBa)** of Spoken (German, English, Japanese) and Written (German) Language, with the following annotation layers: tokenization, lemmatization, morphology, part-of-speech, syntax (constituency and dependency), word senses, anaphora, named entity classification, and discourse relations.

Marie Hinrichs

Research Scientist
Tübingen University, Germany
marie.hinrichs@uni-tuebingen.de

Marie Hinrichs has a B.S. in Computer Science from the Ohio State University. She is currently working on the **CLARIN-D** project in the Department of Computational Linguistics at the University of Tübingen, Germany. She has a leading role in the development of **WebLicht**, an environment for the construction and execution of NLP processing chains. She has also recently become involved in the technical aspects of maintenance and release of the **TüBa-D/Z**, a treebank derived from German newspaper articles. In addition, she is involved in developing a generic execution environment for executing workflows close to the data in the European infrastructure project **EUDAT**.

Nancy Ide

Chair & Professor
Department of Computer Science
Vassar College, United States
ide@cs.vassar.edu http://www.cs.vassar.edu/~ide

Professor Nancy Ide has worked in the field of computational linguistics for over 30 years and made significant contributions to research in word sense disambiguation, computational lexicography, discourse analysis, and the use of semantic web technologies for language data. She has been involved in the development of annotation standards throughout her career, first as the founder of the **Text Encoding Initiative (TEI)**, the first major standard for representing electronic language data. She later developed the **XML Corpus Encoding Standard (XCES)** and, most recently, the **ISO LAF/GrAF** representation format for linguistically annotated data. She is the convener of ISO TC 37 SC4 WG1 on Basic Mechanisms for Language Resource Management, and has participated in the development of several ISO standards for language data. Professor Ide has managed the development of several major linguistically-annotated corpora, including the EU-funded **MULTEXT** and **MULTEXT-EAST** corpora, and, more recently, the **Open American National Corpus (OANC)** and the **Manually Annotated Sub-Corpus (MASC)**. She has been a pioneer in efforts toward open data and resources, publishing the OANC and MASC as the first linguistically-annotated corpora freely available for any use. She is Co-Editor-in-Chief of the journal Language Resources and Evaluation and Editor of the Springer book series Text, Speech, and Language Technology. She has been the Principal Investigator (PI) or co-PI on multiple major US National Science Foundation and EU-funded projects; currently, she is co-PI of the **LAPPS Grid** project, in which context she is developing standards for web service exchange of linguistically-annotated data.

Michael Kipp

Professor
Department of Computer Science
Augsburg University of Applied Sciences, Germany
michael.kipp@hs-augsburg.de http://michaelkipp.de/j15/index.php?lang=en

Professor Michael Kipp is full professor at Augsburg University of Applied Sciences, Germany. Before he was head of the Embodied Agents research group at the Cluster of Excellence "Multimodal Computing and Interaction" at Saarland University and a senior researcher at the German Research Center of AI. He has co-authored more than 70 peer-reviewed publications in the areas of human-computer interaction, virtual characters, multimodality research and video annotation. He created the **ANVIL** video annotation for his research on the automatic synthesis of co-verbal gestures and has since then further developed it with the help of his students, especially in the direction of motion capture visualization.

Brian MacWhinney

Professor
Department of Psychology
Carnegie Mellon University, United States
macw@cmu.edu http://talkbank.org

Brian MacWhinney is Professor of Psychology, Computational Linguistics, and Modern Languages at Carnegie Mellon University. He has developed a model of first and second language processing and acquisition based on competition between item-based patterns. In 1984, he and Catherine Snow co-founded the **CHILDES** (Child Language Data Exchange System) Project for the computational study of child language transcript data. He is now extending this system to six additional research areas in the form of the **TalkBank** Project. MacWhinney's recent work includes studies of online learning of second language vocabulary and grammar, neural network modeling of lexical development, fMRI studies of children with focal brain lesions, and ERP studies of between-language competition. He is also exploring the role of grammatical constructions in the marking of perspective shifting and the construction of mental models in scientific reasoning. In the area of annotation, his work has focused on the enrichment of the CHILDES and TalkBank corpora with phonological, morphological, and syntactic coding. His

research group has also developed coding systems for gesture, speech acts, sign language, and other areas, as specified in the manual for the **CHAT** transcription and coding system. They have also developed methods for converting between CHAT format and 8 other transcript formats, based on a detailed XML Schema for CHAT that includes the format used by the **Phon** program for detailed phonological analysis. The CHILDES database includes corpora from 30 languages and we have developed morphological and syntactic taggers for 8 of these languages. He is particularly interested in developing methods for increased linkage of the CHAT format to other annotation tools.

Diana Maynard

Research Fellow
Department of Computer Science
University of Sheffield, United Kingdom
d.maynard@sheffield.ac.uk http://gate.ac.uk

Dr. Diana Maynard has been a Research Fellow at the University of Sheffield, UK since February 2000, after receiving the Ph.D. in Natural Language Processing from Manchester Metropolitan University. Her main interests are in information extraction, opinion mining, social media analysis, terminology and semantic web technologies. She is the chief developer of Sheffield University's open-source multilingual Information Extraction tools, and currently leads the work on Information Extraction and Opinion Mining on the EU DecarboNet project. Dr. Maynard is a senior member of the current **GATE** team of 12 researchers, and has been involved in developing the GATE architecture and toolkit since its inception (in its current format) in 2000 and has been heavily involved in the design of both manual and automatic annotation tools, in particular from the user point of view. She developed many of the linguistic processing resources in GATE, in particular the core Information Extraction system **ANNIE**, and was responsible for the development of many of the multilingual tools in GATE. She led the GATE team's work on the NIST ACE evaluations, and on the TIDES Surprise Language Evaluation, both of which met with great success and led to the development of tools for a variety of new languages in GATE (Arabic, Chinese, Cebuano and Hindi) in addition to the English components. Currently, Dr. Maynard is best known for her work on opinion mining and sentiment analysis (and particularly for some recent work on sarcasm detection) and more generally for her work on social media analysis (for example, adapting core IE tools to deal with Twitter and other noisy data).

Eric Nyberg

Professor
School of Computer Science
Carnegie Mellon University, United States
ehn@cs.cmu.edu http://www.cs.cmu.edu/~ehn

Professor Eric Nyberg was a member of the **Unstructured Information Management Architecture (UIMA)** steering committee and has many years of experience with annotation systems for information extraction, semantic retrieval, and question answering, including work on annotation type systems in the IARPA AQUAINT, DARPA GALE and DARPA MRP programs.

George Petasis

Researcher
Institute of Informatics and Telecommunications
NCSR Demokritos, Greece
petasis@iit.demokritos.gr http://www.ellogon.org/petasis/

Dr. Georgios Petasis holds a Ph.D. in Computer Science from University of Athens on machine learning for natural language processing. His research interests lie in the areas of natural language processing, knowledge representation and machine learning, including information extraction, ontology learning, linguistic resources, grammatical inference, speech synthesis and natural language processing infrastructures. He is the author of the **Ellogon** natural language engineering platform. He is a member of the program committees of several international conferences and he has been involved in more than 15 European and national research projects. As a visiting professor at University

of Patras he has taught both undergraduate and postgraduate courses. His work has been published in more than 50 international journal, conferences and books. He is the treasurer and a member of the board of the Greek Artificial Intelligence Society (EETN). Finally, he is co-founder of "Intellitech," a Greek company specializing in natural language processing.

James Pustejovsky

Professor & Chair
Department of Computer Science
Brandeis University, United States
jamesp@cs.brandeis.edu http://jamespusto.com

Professor James in the TJX Feldberg Professor of Computer Science at Brandeis University, and chair of both the Linguistics Program and the Computational Linguistics MA Program. He first started working seriously on annotation in 2002, when he led the creation and development of **TimeML** and **TimeBank**, in the context of a six-month IARPA (ARDA) workshop. This was then incorporated into **ISO-TimeML**, which has been adopted as an ISO standard. His involvement with ISO began in 2006, where he is the sub-chair responsible for **SC 4** within **TC 37**. In 2008, he started the working group on spatial annotation in language, which has been adopted recently as the ISO standard, **ISOspace**. TimeML has been used as the reference annotation specification, and TimeBank the reference gold standard for all of the SemEval shared task challenges, TempEval and their affiliates. Likewise, this year at SemEval 2015, SpaceBank and ISOspace were adopted as the corpus and associated annotation standard for Task 8, SpaceEval. In 2012, Professor Pustejovsky and Amber Stubbs released an O'Reilly book on "Natural Language Annotation for Machine Learning", which is intended as a guide to the ins and outs of MATTER, the development cycle for modeling, annotating, training, and testing with ML algorithms. He is, along with Nancy Ide, finishing up a comprehensive "**Handbook of Natural Language Annotation**", to be published later this year by Springer. Other annotation projects he has been involved with include: factuality and veridicality (**FactBank**), with Roser Sauri; semantics of images (ImageML), with Julia Bosque Gil; and temporal annotation for the clinical domain (**THYME**), with Guergana Savova and Martha Palmer.

Anna Rumshisky

Assistant Professor
Department of Computer Science
University of Massachusetts at Lowell, United States
arum@cs.uml.edu http://www.cs.uml.edu/~arum/

Professor Anna Rumshisky received her Ph.D. in Computer Science from Brandeis University in 2009, followed by postdoctoral training at the MIT Computer Science and Artificial Intelligence Lab. Her research primarily concerns natural language processing applications in clinical informatics, computational lexical semantics, temporal reasoning, and digital humanities and social science. She has been directly involved in several large-scale annotation initiatives, including **Corpus Pattern Analysis** and **TimeML**. She has co-organized 2012 **i2b2** Workshop on Challenges in NLP for Clinical Data, overseeing the development and release of the first large-scale corpus of temporally annotated narrative provider notes. She co-developed **Generative Lexicon Markup Language (GLML)** for the annotation of compositional operations in text and co-organized SemEval-2010 Task 7 on Argument Selection and Coercion, based on GLML. She has developed methods for decomposition of complex annotation tasks into pairwise similarity judgments tasks for Amazon Mechanical Turk. She presented this methodology in a 2014 tutorial on Deep Semantic Annotation with Shallow Methods given at the International Conference on Lexical Resources and Evaluation.

Gary Simons

Chief Research Officer
SIL International, United States
gary_simons@sil.org http://www.sil.org/~simonsg

Gary F. Simons is currently the Chief Research Officer for SIL International in Dallas, Texas and Executive Editor of the Ethnologue (http://www.ethnologue.com). He has contributed to the development of cyberinfrastructure for linguistics as co-founder of the **Open Language Archives Community** (http://www.language-archives.org), co-developer of the ISO 639-3 standard of three-letter identifiers for the known languages of the world, and co-developer of linguistic markup for the **Text Encoding Initiative (TEI)**. He was formerly Director of Academic Computing for SIL International in which role he oversaw the development of linguistic annotation tools like **IT (Interlinear Text Processor)**, **CELLAR (Computing Environment for Linguistic, Literary, and Anthropological Computing)**, **LinguaLinks**, and the beginnings of **FLEx (FieldWorks Language Explorer)**. He holds a Ph.D. in general linguistics (with minor emphases in computer science and classics) from Cornell University.

Han Sloetjes

Software Developer
Max Planck Institute for Psycholinguistics, The Netherlands
han.sloetjes@mpi.nl http://www.mpi.nl/people/sloetjes-han

Han Sloetjes has been working as a software developer at the Max Planck Institute for Psycholinguistics since 2003. In 2004 I joined the group of developers working on the multimedia annotation tool **ELAN**, a tool that emerged sometime at the end of the nineties. Between 2006 and 2007 he became the main developer responsible for maintaining and extending ELAN, at which point he also became the main developer providing support and training.

Brett South

Research Scientist
Department of Biomedical Informatics
University of Utah, United States
brett.south@hsc.utah.edu http://medicine.utah.edu/bmi/people/staff-technical/index.php

Dr. Brett South has extensive experience leading annotation projects and manual human review efforts that support development of clinical NLP systems. He has 18 years of academic and professional experience in various research, leadership and operational roles. He is currently a Senior Scientist and Post-doc working under the primary mentorship of Professor Wendy Chapman in the Department of Biomedical Informatics, University of Utah. Previously he was a Senior NLP Research Engineer for the **Nuance Clinical Language Understanding Group** where he helped lead a group of 80 clinical language analysts tasked with large-scale semantic annotation of clinical corpora to support development of a computer-assisted coding module. Previously he was with the Division of Epidemiology and VA Salt Lake City IDEAS Center as a Senior Research Scientist serving in several investigative and co-investigative roles on the VA CHIR/VINCI collaborations. Prior to completing his Ph.D. in Biomedical Informatics from University of Utah, Mr. South obtained his Master's degree in Health System's Management from University of Maryland, Baltimore. His research interests include: clinical NLP, integrating efficiencies with manual human review tasks via improvements in tools, workflow modifications, or distributed review, human cognition, and data analysis.

Pontus Stenetorp

Researcher
University of Tokyo, Japan
pontus@stenetorp.se http://pontus.stenetorp.se

Dr. Pontus Stenetorp is a post-doctoral researcher in Natural Language Processing. He is mainly interested in Natural Language Processing and Machine Learning, more specifically, parsing, information extraction, annotation tooling, deep learning, and representation learning. He is co-creator of the annotation tool **Brat**.

Stephanie Strassel

Senior Associate Director
Linguistic Data Consortium
University of Pennsylvania, United States
strassel@ldc.upenn.edu https://www.ldc.upenn.edu

Stephanie Strassel oversees the Linguistic Data Consortium's Annotation and Collection Groups and is responsible for directing all aspects of data collection and creation, human subject research and linguistic annotation for a diverse set of externally sponsored human language technology research programs and evaluation campaigns. She has acted as PI or Co-PI on multiple efforts including most recently DARPA DEFT, BOLT, RATS, MADCAT, GALE and Machine Reading; IARPA ALADDIN; NIST OpenMT, LRE, SRE, OpenHaRT, TAC KBP, Rich Transcription, VAST and HAVIC. Previous projects include **TIDES**, **EARS**, **ACE**, **Phanotics** and **TRECVid**. In her work on sponsored projects Ms. Strassel has overseen all aspects of corpus creation including linguistic annotation for a diverse set of technologies including machine translation, speech recognition, question answering, handwriting recognition, document classification, topic detection, information extraction, knowledge base population, natural language understanding, speaker identification, dialect recognition, multimodal event detection and related areas. To date Ms. Strassel has co-authored over 100 corpora published in LDC's catalog, with several dozen additional corpora pending publication. Ms. Strassel has experience with virtually every type of linguistic annotation and has led LDC efforts to define dozens of new annotation and data collection protocols, many of which are widely used in resource creation elsewhere. She has experience with resource creation and annotation for over 40 linguistic varieties including several low resource languages. At LDC Ms. Strassel directs a staff of 13 full-time and 75+ part-time linguists, managers, researchers and programmers and maintains a large network of independent contractors for a variety of languages. She provides leadership for LDC senior staff, developing and disseminating infrastructure for activities that support multiple functional areas. She frequently serves as an external expert on language resource creation activities via review panels and oversight committees.

Marc Verhagen

Senior Research Scientist
Department of Computer Science
Brandeis University, United States
marc@cs.brandeis.edu http://www.cs.brandeis.edu/~marc

Dr. Marc Verhagen is a Senior Research Scientist at Brandeis University, prior to which he was co-founder and master toolbuilder at **LingoMotors**. He holds a Ph.D. in computer science, an M.A. in computational Linguistics and a B.A. in Geography. Recently, he has worked on temporal and spatial processing, relation extraction from PubMed abstracts, technology extraction from technical texts, and interoperable web services for Natural Language Processing. Verhagen is the main developer of the **Tarsqi Toolkit** for temporal processing and the creator of various simple annotation tools as well as the web-based **Brandeis Annotation Tool (BAT)**. He participated in the creation of the **TimeML** and **ISO-Space** annotation languages. In the context of organizing the **TempEval-1** shared task, he oversaw annotation of the task data, creating ad hoc annotation tools in the process.

C. Pre-Workshop Survey Questions

The following survey was distributed to the workshop participants two weeks before the meeting. The survey was designed to elicit information relevant to the workshop topic, as well as stimulate participants to begin thinking about the issues involved.

8.1. Welcome Message

Welcome to the Pre-Workshop Questionnaire for the NSF-Sponsored Workshop on Unified Annotation Tooling! This questionnaire collects information about your experiences with developing and working with annotation tooling, and well as with the process of annotation itself. Your answers are critical to an effective and productive workshop, as well as to later research efforts based on the workshop's recommendations. Please be as verbose, clear, and complete as possible in your responses. I anticipate this questionnaire will take 1-2 hours to complete. Please complete it by Wednesday, March 25, 2015 (next week), so that I have enough time to analyze the results and return them to the participants before the workshop begins. When filling it out, please don't look at any of the materials I have sent; this will help us maintain our collective creativity and avoid getting stuck in topical ruts. Throughout the questionnaire I refer to "UAT" (without an article), which should be read as "Unified Annotation Tooling". "UAT" is a catch-all term to describe any approach which seeks to solve the problem of inadequate, inconsistent, and incompatible tool support for annotation. I make no commitments in this questionnaire about whether UAT will be one tool or many; developed or supported by one group, country or field; or whether it will be a new tool, an improvement of existing tool, or somewhere in between. I very much appreciate the time you are taking to answer these questions and attend the workshop. Your participation is extremely valuable!

Sincerely,

Mark

8.2. Basic Information

8.2.1. Biosketch

Please provide a short biography of yourself suitable for use on the workshop webpage and in the NSF report. Point out your key work in annotation, including corpora you have helped create, projects you have directed, tools you have written, and so forth. Make sure to indicate explicitly what you are known for.

8.2.2. Website

Please provide your preferred webpage for linking on the workshop website.

8.2.3. Affiliated Fields

List the fields or disciplines with which you are most closely affiliated.

8.2.4. Venues

Give a sample of the most important conferences and journals where you normally publish.

8.2.5. Quoting Questionnaire

May I quote your responses to this questionnaire in future reports and proposals?

8.2.6. Quoting Discussion

The discussions at the workshop will be recorded to facilitate capturing and summarizing the recommendations. May I quote your statements from the discussions in future reports and proposals?

8.3. Motivations & Scope

8.3.1. Motivations

Give your top 3 reasons for the creation of UAT, in order of importance.

8.3.2. More Motivation

Are there any additional motivations that you think are very important, beyond the first three?

8.3.3. Disciplinary Scope

Many fields and sub-fields use annotation or annotation-like approaches in their work. While we probably cannot create a tool that is all things to all people, we can strive to achieve a relatively comprehensive solution. Which fields, sub-fields, and disciplines should have their needs prioritized when designing and implementing UAT and why?

8.4. Annotation Tools

8.4.1. Tools Used

List the tools that you have used in your annotation work. Note where you created tools to fill a certain gap (explain the gap).

8.4.2. Tool Problems

Thinking back to your experience with existing annotation tools (other than tools you built yourself), what are the top 3 problems you encountered that prevented you from using those tools for your work?

8.4.3. Additional Problems

Are there any additional problems you encountered that you think are very important, beyond the first three?

8.4.4. Best Tools

What do you think are the most functional and useful annotation tools available for your work, and why?

8.5. Workflows

8.5.1. Workflows Used

Thinking back on any annotation projects you have managed, describe the workflow(s) you have used to create annotated corpora. By "workflow" I mean the step-by-step procedure used to go from nothing to an annotated corpus that is downloaded to a user's machine. Provide every step you can think of from start to finish, especially steps involving any interaction with a computer. Be as detailed and complete as possible. More detail is better than less.

8.5.2. Workflows Prevented

What workflows have you considered using in past projects, but found the available tooling inadequate, causing you to change your project design?

8.5.3. Important Workflows

Aside from the types of workflows you have mentioned above, are there any additional workflows you think should be supported by a UAT?

8.5.4. Workflow-priority

Please prioritize in a ranked list all the workflows you have mentioned so far.

8.6. Annotation Schemes

8.6.1. Schemes Used

List as many annotation schemes you have used in your work as you can. Give a small amount of detail to disambiguate the scheme from others with which it may be confused.

8.6.2. Schemes Prevented

What annotation schemes have you considered using in past projects, but found the available tooling inadequate, causing you to change your project design?

8.6.3. Important Schemes

Aside from the types of annotation schemes you have mentioned above, what sorts of file formats or annotation schemes do you think absolutely must be supported by a UAT to be of broad use?

8.6.4. Scheme Priority

To the extent possible or meaningful, please prioritize all the annotation schemes you have mentioned so far. If there are too many schemes, concatenate them into the last entry in an ordered list.

8.7. Management & the Future

8.7.1. Implementation

UAT might be designed and built from scratch, might be a modification of an existing tool, or might take a hybrid approach, leveraging certain parts of existing infrastructure while adding completely new parts. Do you have opinions on the feasibility or preferability of these various approaches?

8.7.2. Management

One pernicious problem is the sustainability of UAT. Funding, engagement of researchers, and developer interest waxes and wanes. How would you propose to manage UAT such that it was supported and developed over a period of, say, 7-10 years?

8.7.3. Collaborations

If the NSF and/or the EU decides to fund an effort to develop UAT, describe what you think you might be willing and able to contribute to such an effort.

8.7.4. Other

Do you have any other comments that you think would be useful, on any topic related to the workshop (even the structure of the workshop itself)?

8.8. End Message

Thank you very much for you answers. I will send out a synopsis of the results of this questionnaire a few days before the workshop, to help workshop participants to get into the right frame of mind. I look forward to seeing you in Sunny Isles! Best regards, Mark

D. Collated List of Survey Answers

The answers to the questions on the survey were collated (indicated by major headings below), and similar answers were grouped (indicated by minor headings). The lists below contain the verbatim answers given by the workshop participants to the survey.

Motivations to Solve the Problem

Achieve Greater Access to Data

- Increased accessibility of annotated data / corpora
- To streamline the process through which "ordinary working linguists" (a term usually used to encompass field linguists with relatively little computational sophistication) can make useful language resources on the basis of data collected in the field.
- To make it easier for the general public (including speaker communities) to make use of language resources created by scholars. (This is of central importance to many endangered language linguists, but less important to me personally.)
- There seems to be a split sometimes between annotating and searching through the annotation, and by keeping the parts more uniform UAT may provide a way to better integrate searching. This is related to my second point, in that flexible UAT would provide different views of the same data to assist in annotating (e.g., sentence-by-sentence, construction-by-construction, annotation inconsistencies, interesting connections between layers, etc.), and such views are likely useful for users who simply want to search.
- Encouraging cross disciplinary interaction. Fact that different disciplines is completely different tools really shuts off a lot of data from a lot of people. There's a lot of scope in opening up the kind of tools that we use to broader use and promoting interoperability of tools for collecting data analyzing data.
- Enable groups who lack substantial computing expertise and resources to better exploit linguistically annotated data
- The need to embed viewing and annotation of text in a variety of different applications

Avoid Reduplication and Reduce Cost

- Reduplication of work across different projects is, of course, the most obvious issue. Stemming from this, though, seems to be a slight discouragement of developing innovative tools because time needs to be spent on getting the core annotations working and thus cannot be spent figuring out ways to improve interfaces, consistency checking, IAA modules, etc.
- All-in-one solution: The re-use of generic infrastructure for e.g. annotator management, agreement computation, and project workflows.
- To make it easier for people to get started in collecting language resources, and annotating in such a way that they remain useful throughout their life. So we can spend less time developing the same old tools and more time actually studying language.
- More effective use of available funds - better return on investment
- sustained maintenance of tools
- Reduce (or better eliminate) the fragmentation caused by the many tools available
- Concentrate the development effort to a single tool, which will result in more capabilities for the tool, robustness, etc.
- Less bugs, as widespread usage will allow easier detection/bug fixing
- Reduce overall cost/time required to produce linguistic resources
- There will always be a need for annotations
- current segmentation of effort
- a “software waste cycle”, in which tools with substantially overlapping functionality are repeatedly created from scratch by researchers and teams all over the world, causes enormous waste of funding and human resources;
- reduction of marginal cost for developing new systems / lowering the barrier to entry by non-developers
- Reducing the labor currently expended on continuously re-creating annotation tools for different annotation teams.

Ease Annotation

- Facilitation of users' workflow, from coding to analysis
- Improved access to annotation tools
- Ease of starting and completing a wide-variety of annotation tasks
- Classification of annotation tasks (annotation objects and annotation work flows).
- To facilitate workflows where different individuals are adding different kinds of annotations to the same primary linguistic data and where these annotations can be straightforwardly integrated to create richer language resources.
- Ability to use a variety of taggers on CHILDES and TalkBank data
- Ability to analyze web pages in a variety of language for second language learners
- Linkage to systems for automatic segmentation, pause detection, and perhaps ASR
- Efficient expansion of annotated corpora that are free of errors, which can be allowed to grow in scale with relatively easy maintenance of the data.
- Efficient presentation of annotation tasks to human annotators, which take advantage of annotator strengths and speed annotation with automatic pre-processing where possible.
- Distributed annotation: The collaborative annotation tool will be used in a distribution fashion with the only requirement being internet connectivity. Annotators can work at any time and from anywhere, without concerns for data losses and continuous intervention to save the data.
- Unlocking a larger workforce: The main goal of an annotation tool is to generate large annotated corpora. Similar to crowdsourcing platforms, it is possible to generate larger amount of annotated corpora more quickly by making them accessible to larger workforce.
- To make it easier for people to get started in collecting language resources, and annotating in such a way that they remain useful throughout their life. So we can spend less time developing the same old tools and more time actually studying language.
- Chaining processing steps from different providers is very attractive to us
- Easy combinability of existing functionality is the key for the average user, with highly sophisticated functionality available in some tools
- For making it Unified: That annotation of under-resourced languages can become a collaborative enterprise involving members of the language community itself
- there is very little support for crowdsourcing annotations; collaborative, distributed annotation; and semi-automatic annotation involving a “human in the loop”, which are becoming increasingly common.
- the advent of linked data is demanding that consistency among annotation models be achieved, in order to ultimately enable the exploitation of all kinds of annotated data in the Semantic Web.
- Promoting awareness of the entire life-cycle / process associated with the creation of type systems, annotations, corpora etc. from completely human linguistic analysis to automatic computer processing

Facilitate Archiving

- To facilitate the archiving and long-term preservation of linguistic annotations. (This is something quite important to me, but it falls out more or less naturally from the other issues listed above. So, I've made it a lower priority here.)
- Improve sustainability situation

Improve the Range of Annotations

- Improving the range of annotation layers that can be easily annotated.
- Multi-language situations being researched pose a specific problem to tools designed to support one (specific) language only. It seems much for feasible to mix small components in as needed than to wait for all the necessary tools to be upgraded and made compatible.
- For annotation tooling: To adequately document and describe the thousands of languages of the world

Increase Quality of Results

- Promoting best practices in annotation
- Encourage replicability/comparison/evaluation of results (like IAA)
- we need means to consistently document annotated data and evaluate annotation quality; a well-conceived infrastructure for annotation can support this

- Creating best practices/methods for visualizing annotations and making annotations quickly and easily
- development of global requirements and shared best practices (processes associated with use of UAT);
- How best practices / requirements from both realms should better inform each other and the creation of UAT
- Establishing best practices for making annotation tasks as straightforward and as cognitively light as possible for the annotators.

Obtain Interoperability

- Interoperability of annotations and tooling
- Greater ease of exchanging annotations (even with the current widely adopted standards)
- To allow all kinds of linguistic data to be more easily and opportunistically used across diverse tools.
- To facilitate the interoperation of language resources developed for low-resource languages, especially those created by field linguists working on endangered languages. (This has been an important aim of many working in the area of digital standards for endangered languages, and it is of interest to me, but lower priority, in personal terms, than the first three points listed above).
- To make it easier for those working in other disciplines to use language data since they would only need to learn one annotation scheme (ideally).
- Interoperability. Annotation resources each have different strengths and weaknesses, as they generally represent different types and levels of detail in their representations. To take advantage of strengths and overcome weaknesses, and to ideally combine independent annotations into a larger, more diverse training corpus, these resources should be interoperable.
- To make sharing a data between different disciplines easier and therefore make resources when they have been created more widely useful.
- Improvement of the decision process for users - trust building will be less difficult - promise of increased interoperability
- ability to process annotations created by other tools
- standardization
- For making it Unified: That the data might interoperate smoothly across a whole ecology of tooling
- Encourage interoperability of linguistic resources
- lack of interoperability
- the development of linguistically-annotated corpora is currently inconsistent; the resulting proliferation of vastly varying annotation models and practices prohibits interoperability and reuse of these resources, which in turn prevents the exploitation of high-quality annotated language data by groups that lack substantial computing expertise and resources; (
- Standardization and interoperability;

Provide Extensibility

- Increased probability of a community process for extending the tool
- Individual tools do not generally account for various possibilities of (future) annotation needs, and UAT would ideally address this by attempting to develop flexible infrastructure which addresses a range of annotation types.
- Enhanced flexibility: A general-purpose annotation tool provides better flexibility, in such a way that any type of annotation layers can be created, depending on the data collection need of the target application.
- Creating a universal set of recommended tools that would be easy to adapt or extend to handle most common (and not-so-common) types of annotation tasks.

What Disciplines Should Be Prioritized?

No Prioritization Needed

- I don't think one can really [prioritize disciplines]. A better approach is probably to "cluster" the various disciplines to see which ones are closer together and which ones are further apart.
- I don't think it makes sense to identify fields and/or disciplines where the tools and procedures should be applicable. I think we need to abstract away from that. What should be identified is what kinds of annotation should be made possible and what kind of work flows are supported.

- If "should have their needs prioritized" means that they should agree on what to forget about, that's difficult from the beginning on. I believe that progress can only be achieved by providing extra value to specific disciplines.
- No single discipline is more identifiable than other, since data markup and annotation of language-related or linguistic information can be a goal within any field. That said, we can measure the linguistic annotation needs of a discipline by asking: how significant a component of the data analytics problem in your field is natural language and text? For humanities and the languages, it is obviously a pressing need, given that is the natural domain of discourse. For the social and behavioral sciences, including economics, annotation can be an enormously critical source of additional metric for trends, sentiment, and other domains. Even in GIS and epidemiology, there are needs involving language annotation.
- I do not feel that any field needs to be prioritized
- I see it not as a matter of priority, but as a matter of classification. The same types of annotation tasks may be required by different subfields, while the same subfield will require different annotation types, depending on the task. In the tutorial we gave at LREC last year, we developed a mini-typology of annotation tasks that classified different types of annotation along several axes, including distinguishing surface spans / attribute annotation from relation annotation, separating annotations with span-specific vs span-dependent label sets, and annotation of overt vs. covert phenomena and elements.

Proffered Prioritizations

- Endangered languages are disappearing rapidly, so a priority for the needs of language documentalists and field linguists would make sense.
- Fields and disciplines which could achieve significant impact (e.g. biomedical QA, legal document analysis) through more effective annotation, corpora creation and automatic processing are obvious targets.
- Documentary linguistics -- Because there are thousands of languages to document and describe, and the general consensus is that half of them are in danger of dying by the end of the century
- social media/blogs; clinical reports; biomedical literature
- Natural Language Processing; 2. Corpus Linguistics; 3. Semantic Web; 4. Language Documentation; 5. Psycholinguistics; 6. Education
- NLP, biomedical text mining, digital humanities, machine translation, social media mining NLP
- computational linguistics and language technology; 2. corpus linguistics; 3. literary studies; 4. History; 5. political science; 6. sociology
- Linguistics, Sociolinguistics, Language Learning
- Semantic annotation; Annotation with ontologies on segments; Sentiment analysis; Argumentation mining
- computational linguistics; corpus linguistics; social sciences (historical, politics, etc.,...)
- Speaking entirely from the perspective of LDC and our sponsored project needs, I see the greatest need in the area of UAT for semantically annotated and lexical resources.

Problems with Existing Tools

Difficult to Learn

- too-steep learning curve, inadequate documentation/tutorials available
- Steep learning curve
- Opacity of functionality and ease of adoption

Difficulty Running or Installing

- Don't run on particular operating systems (e.g., OSX)
- Often what they do is so simple that we just reimplement them in CLAN.
- Difficulty accessing and running the tools. Some annotation tools that I've worked with require annotators, who may not be familiar with a Unix environment, to launch their tools within a Unix environment by giving a command. Although they are trained on this process, this can be a frustrating barrier to annotators who may have difficulty navigating a Unix environment and giving the correct command.
- Difficulties installing and setting up tools, and finding out how they work and what they do.
- Installation complexity
- tool requires computing environment we can't easily support (e.g. special server, security concerns)

Inadequate Importing, Exporting, or Conversion

- Difficulties in going from the annotation tool to data formatted for publications. Linguistics publications tend to require data in a visually appealing interlinear-glossed format. Most of the tools that I use don't make it easier to explore such a format. So, to make a publication, one has to do a lot of the work by hand. Sometimes, rather than annotate properly, I just start with the publication-ready format.
- Only partial support for conversion between annotation formats
- Lack of a standardized data interchange format that would allow the new tool to interoperate nicely with the other tools I was using
- data output format is too inflexible and/or doesn't meet our requirements
- "Academic code" and built-in platform assumptions
- lack of means to input/output annotations in a desired format
- no ability to read in or perform pre-annotations
- tools don't always handle schemas with a large number of types in a graceful manner
- mapping between type systems created by different tools / schemas
- They mostly failed to interoperate with CHAT format.

Lack of Documentation or Support

- Support for the tool
- For syntactic dependency taggers, the problem has been the poor documentation of the range of tags, bad interoperability, etc.
- some tools lack enough documentation to know exactly what to do
- Problems with maintenance of tools, finding tools that look useful but that nobody is taking care of
- Lack of documentation, tutorials, examples that were adequate to figure out the tool within the timeframe I was willing to spend on evaluating it
- Lack of documentation, at best, a text file. It is difficult to even get the tool running.
- Lack of support, people leave academia and suddenly there is no one to ask about how things work.
- lack of consistency of formats and scheme design;

Missing Functionality

- Limited functionality: no “coding scheme”
- Often the tool only provides a subset of the features desired.
- Lack of support for languages written where diacritic marks are used to encode tonal contrasts. This is a concern that is somewhat technical but, at its heart, relates to how some transcription systems are based around a model where "accents" encode tonal contrasts, but no tool that I know of is aware of this, which causes problems for automated parsing and searching. (I can give more details if you'd like. This is a particular problem for Africanists.)
- For morphological analysis, there are not really any tools out there that get to the level we need. The Xerox/PARC FST framework has computational problems and is not open enough for general use.
- Lack of easy search functionality for "homegrown" types of searches, i.e., searches that are specific to our particular needs. For example, we wanted to find mismatches between annotation layers that might have helped reveal something about the annotation scheme we would have wanted to revise.
- Visualization is thus sometimes a challenge
- while dedicated video and image annotation tools exist, their support for text annotation is poor.
- Many lack support for collaborative annotation over the web, which is now a key requirement for the projects we work on
- Lack of sufficient flexibility in supporting different annotation flows and mixing with automatic bootstrapping and adjudication methods
- Many tools handle video now, but don't give it a central role
- many tools target either single researchers or languages with an established writing system with an orthography, both are not the case for our research.
- The tool was missing key functionality I was looking for
- lack of workflow support (inability to manage users & assignments, track progress and perform other management functions)
- lack of some key functionality

- lack of means for quality control and assessment
- can't be embedded in other applications - is a standalone
- not able to do instance-level annotation as well as document- or corpus-level annotations
- cannot create shortcuts for users (e.g., mark all instances of "pain" as Symptom - don't make me re-annotate the same words over and over."
- Lack of necessary functionality for the task at hand.
- Lack of “coverage” e.g. inability to handle relations between either overt or covert entities.

Poor User Interface

- Bad user interface: you want coding to be efficient because it is very time-consuming (and boring!)
- My main beef, including on tools I wrote, is usually the ease of use of the interface.
- Tediousness of performing some tasks (For example, the Alembic Workbench required a long sequence of click to add a relation.)
- General usability. Most tools produced by scholars have clunky user interfaces and, in particular, they don't follow the standards used by professionally-made software. This requires learning new interfaces and workflows for each of these tools, and I choose carefully when it is worth learning them.
- Usability
- Annotator fatigue
- poor design makes things cumbersome or error-prone or unnecessarily tedious for annotators
- too many clicks - burdensome to use
- Uncomfortable, difficult, or needlessly silly interface.
- Too much functionality, too much configurability (MATE workbench)
- Complexity and lack of transparency. Some annotation tools benefit the annotation process by proving built-in reference tools. While this is advantageous, it can also create an interface that is intimidating to annotators, especially when the functionality of the tool is not totally transparent to the annotator
- Poor workflow design, i.e. sequences of actions required to create an annotation were too long and/or seemingly unmotivated.

Problems with Extensibility

- I found integrating rhetorical relations difficult in ANVIL.
- Lack of customization of data formats and annotation categories
- Rigid restrictions on annotation types - e.g., not (easily) allowing for two separate part-of-speech fields.
- Many tools were annotation task specific and/or closed source, which made them hard to modify and customize to new corpus annotation needs/tasks
- Inability to adapt tools to established practices in sub-fields
- Customization of annotation schemas
- too difficult/time-consuming to modify compared to building our own custom tool
- "Academic code" and built-in platform assumptions
- difficult to customize
- Workflows that allow for partial annotation or annotation of one "layer" or component of a problem have been hard to develop.

Unstable, Slow, or Buggy

- Stability and reliability of the tool.
- Speed. Ideally, the annotation tool should work as quickly as the annotator. Some web-based tools and those accessed via a secure shell can become very slow outside of the ideal connectivity contexts.
- Tools with bugs. We cannot take for granted that a tool simply saves data properly, instead of crashing or saving data with small errors.
- Some tools are not well-suited for large data files,
- tool is buggy
- Bugs. The more complicated the annotation task and the corresponding tool, the more likely one finds bugs in the way the tool functions.

Desired Capabilities

Import/Export Capabilities

- good export functionality is a must: easy to export such that the dada can be used in a publication
- JSON needs to be supported for handling tweets
- Both inline and standoff markup needs to be supported
- Reading any text format and XML would be a necessary capability.
- importing and exporting data and annotations
- transferring between formats

Media to Support

- core ability to manipulate time-aligned data (heavy data, such as multi-HD video streams)
- ability to annotate data that cannot easily be moved to different places because of bandwidth limitations or legal reasons
- Support for images and, even possibly, video annotation, when mixed with text: social media and other digital content is no longer just textual and while dedicated video and image annotation tools exist, their support for text annotation is poor. Ideally, we want something unified and capable of doing all those to at least some degree.
- It is expected that in many fields the use of and availability of multimedia recordings will increase

Supported Schemes

- should provide tools to do syntactic and semantic role annotation in many different languages
- Providing information on coreference should also be supported by UAT. Coreference information can span documents, and therefore a tool that allows users to visualize coreference chains, and connect entity mentions, can be difficult to develop.
- semantic annotations need to be enriched with causal/temporal chains that go beyond a single clause or sentence.
- I think it will be important for a UAT to support some sort of Linked Data friendly format (e.g., RDF)
- UAT should support a tier-based, multilevel, hierarchical annotation scheme. And probably a syntax or treebank type of scheme as well.
- The file format should be XML based.
- linking transcripts to audio
- Support for relevant linguistic annotation standards, such as ISO/TC 37/SC 4
- Support as many possible import/export formats as possible.
- it is clear that annotation schemes is not a closed set, so it is important that UAT be defined in terms of extensible schemes that can represent new schemes (such as RDF and the Linked Data framework).

Workflows

- Creation of dependency parses - a web-based, fast, user-friendly tool to create dependency parses for a variety of languages would be very useful, providing syntactic information and semantic dependencies for languages that may not be suited to a phrase structure analysis.
- I believe that it would be good to support more "distributed" annotation, in an "assembly line" process where people with different skills perform different kinds of annotation.
- work flows should be customizable
- there needs to be support for crowdsourcing and distributed (collaborative) annotation and annotation involving “human-in-the-loop”.
- Multi-role support, including user groups, access privileges, annotator training, quality control, and corresponding user interfaces.
- Shared, efficient data storage to store and access text corpora and annotations.
- Support for automatic pre-annotation services and their configuration, to help achieve time and cost savings.
- Flexible workflow engine to model complex annotation methodologies and interactions.
- I think it would be fine if an UAT just does the annotation part of the workflow. Collection of raw data and (possibly) conversion to usable formats, project management and metadata creation, uploading to an archiving /

long term storage system: those parts might best be dealt with by other tools. Where possible smooth integration with tools that perform these other tasks should be provided.
- Distributed annotation approaches
- A workflow that incorporated interactive annotation in a way that was scalable to larger corpora would be good to investigate
- Every successful annotation project has been iterative. [support for this is a must.]
- possibly interactions with MTurk
- we need a very flexible and powerful workflow system that allows us to arbitrarily combine manual and automatic annotation stages, to manage data sets and user roles, and to prioritize the various components
- supporting multiple annotators
- adjudication
- assignment of annotators to annotation tasks
- ability for annotators to work online and offline
- A UAT should be as versatile as possible: offer a set of workflows, but not impose them, allowing the annotation within a custom workflow
- We can enumerate annotation tasks. Workflow is the general problem of how we orchestrate the flow in which the output of one task is the input to the next. What is needed is not a set of workflows that are supported, but a system for orchestrating workflow among tasks

Miscellaneous Functionality
- the tool should be designed to work on all languages, especially low-resource ones, from the start, rather than a more usual approach where a tool is designed to work on well-known languages first, with other languages only supported after
- Open source: An open source annotation tool can be extended with new functionalities, and is thus subject to a collaborative (programming) process. This flexibility makes a tool more attractive for people that conduct and oversee annotation projects.
- Annotator analysis (IAA, agreement with adjudicator, speed)
- Built-in logging and commenting
- It should include a web-based version.
- Annotation tools (or composers of multiple tools) should facilitate different layers of annotation and abstraction, in an easy to use and simple to understand fashion. This layered annotation should be easily integrated or composed with other levels and domains of annotation.
- The domain user or expert (e.g., clinical or legal texts) should have a gentle learning curve and an intuitive experience for annotating the material in text that they are most familiar with.
- An annotation tool should be integratable on top of any document format (doc, pdf, txt), and any basic application program; e.g., MS Word, Adobe, Excel, iPhoto, Latex, etc.
- Keeping the learning curve for tools very low [from the point of view of the annotator] is critically important.
- UAT should be modular in design and implementation.

How to Implement the Solution?

Build Links and Reuse Existing
- In my opinion the most promising avenue is to build bridges between tools, either in the form of additional software or in the form of tool extensions and/or in the form of interface and file format conventions that people agree to. I am not speaking of an all-encompassing super file format though. We tried this once by using Annotation Graphs and it turned out that a file format always has limitations with respect to a tool which have to be fixed with special notation which in turn makes a unifying format meaningless
- I want methods for linking existing tools.
- I think that we should focus on creating a web-based application that allows users to easily access a variety of different annotation tools that can be used in a modular fashion to efficiently complete the desired steps of annotation with consistent input/output formats that ensure interoperability of annotated corpora.
- Maximal compatibility with existing major tools is going to be one of the biggest problems. This should perhaps be the primary concern when designing a new tool.

- Starting from scratch always has the danger of repeating all the mistakes that existing tools have already made and fixed.
- Also, how can one claim to unify annotation tooling while creating yet another tool? Will it be better than the old ones? And most of all: when will it be ready for use?
- It should, wherever possible, re-use existing tools and workflow engines.
- I probably lean towards trying to leverage as much as possible
- I think it is important to build on what we have now that is working well. Previous efforts to build the one true tool have not worked but there are many tools that have a solid following and support base. Establishing new tools is hard.
- My feeling is that we need to make use of the emerging interoperability standards and look at how to adopt existing tools with a good following to work with them and so work with each other.
- Building on existing tools has the advantage of potentially being more cost effective because of the re-use of existing implementations. But adopting existing code might appear to be hard and more time consuming than anticipated.
- Existing best practices must be exploited and reused.
- Ideally, code that works as expected (and is expandable) must be reused as much as possible.
- I would suggest leveraging an existing generic infrastructure (e.g., the LAPPS framework), providing support for annotation with existing annotation infrastructures such as GATE and UIMA, and, for other purposes, building on and/or adapting existing tools such as LDC's WebAnn and Darmstadt's WebAnno. This would include customized interfaces for different users/applications, and even the ability for a user to specify and control the features of the interface.
- I strongly suggest that any such tool should at first be an integration or modification of current tools.
- The best of present tools should be taken as a "must have" for any redesign into a new functional specification for a new tool.
- There is something to be said though for selecting a popular tool and tinker with it till it better fits UAT principles.

Difficulty in Reuse

- Extending an existing tool can be quite painful given the evolutionary growth of most tools and the ensuing chaos in the code base
- I generally don't ever favor a "build from scratch" approach, but that doesn't mean it's not the right approach here. I don't think any existing tool is good enough to form the basis for UAT.
- Design and build from scratch has many advantages; start with a clean slate with no legacy code that might not be flawless. It might be the most expensive option, re-implementing functionality that has already been implemented by others many times
- Especially when attempting to build on multiple tools the question of supported platform(s) and applied programming language(s) becomes manifest.
- With all kinds of OS dependencies, some fresh start might make life much easier.
- While it is often a good idea to re-use code, sometimes it is more practical to start over.
- I always like the idea of building on what has already been done, but one reason there are so many tools is that existing tools aren't usually easily adapted to meet the needs of another project.

No One Tool is Possible

- Best to avoid something like GATE by focusing on interoperability.
- I think we need to view UAT not as an ideal tool to be developed, but as an ecology of tools that interoperate over common formats and schemes.
- I suspect that building an all-encompassing single tool from scratch would be doomed from the start. Clearly we have a variety of groups that are willing to create annotation tools, the problem lies in the lack of an incentive to work together and to create tools that work well with each other. Thus it would most likely be a better idea to somehow coordinate these groups. Perhaps towards a unified API of some sort?
- I do not believe there can be a UNIVERSAL tool, and it is difficult to address this question without narrowing the types and usages of the annotations that are potentially being produced.

- I do not think there is any likelihood of a single annotation tool for all tasks and domains, it should be the case that all major annotation tools or environments should be interoperable, both in functionality and in content produced.

Miscellaneous

- My hunch would be the hybrid approach. It is important to incorporate formats and other elements of existing workflows, whether embodied in a tool or not. But I think the core would be implemented from scratch.
- What I've always thought would be a good approach, though admittedly I am not a developer, is one where the core of a UAT would be a kind of "engine" (adapted from the notion of a browser engine) designed to perform basic operations over a standardized annotation model (e.g., Annotation Graphs, just to pick one) such as "add annotation", "delete annotation", "modify annotation", with different projects building new tools on top of this engine.
- More importantly, maybe, is the decision on what kind of devices the UAT should run and, related to that, whether it should be an online or offline tool (or both). It will not always be possible to port a desktop application to an app for tablet or phone (or other new devices of the future). The decision on this might determine whether it is at all possible to build on existing tools, or at least on which ones of them.
- Re-engineering with maintainability in mind might be a good solution. This of course does mean that everything needs to be reinvented.
- WebLicht gives online access to annotation tools, eliminating the need to download and install the tools. It uses a building-block approach, which allows users to mix-and-match tools for different annotation layers. This approach is made possible by using a common data exchange format, so that one tool can process the output of another tool
- As I mentioned, I think we need a set of tools, specialized for different (general) types of annotation tasks - cf. the mini-taxonomy of annotation tasks.

How to Manage/Fund/Organize the Solution?

Funding

- If it is possible to acquire funds for a longer period of time to guarantee development, maintenance and support for a longer period of time, that should be preferred, of course, but I don't know of such funds.
- At least with DFG, the German Science Foundation, it has become possible to get support for creating "research infrastructure". Having such an initial grant to set up a project in a way that it can be maintained by the user community seems one way to go.
- I would imagine funding coming from two sources: -core infrastructure and sustained maintenance via NSF CRI or similar programs -customized "modules" for particular annotation tasks via individual project funding Any group who wanted to access the core infrastructure & services could be required to contribute funding to customize the module(s) that would support their particular task(s).
- subscription services can bring in some amount of support. The problem with this is how to bill, and how to differentiate users. Ideally, a platform or tool should be as freely available as possible, so as to encourage adoption and use by as many people as possible. Silver and Gold levels of support might be considered for corporate sponsors, once they see a monetizable payoff.
- I would propose to NSF (or whoever), a project that would work on a UAT as one component and also develop a sustainability model as another component. This latter component would involve workshops, expert consultation, etc., and, by the end of, say, two years, I'd hope for a proposal which could be tested. In other words, I'd treat sustainability as a research question like any other. I'd also ask around for what disciplines have pulled this sort of thing off, probably starting with biology, since my impression is that they have tools which have been supported for a while (though this is helped by the fact that they have a lot of money).
- By building a tool that will attract a community that understands how to extend it. If the community needs the tool, it may provide resources to help sustain the tool.

Implementation

- having an excellent method of distribution of code & versions is also critical to getting it going.
- To me, simplicity and ease of use is often the thing that puts me in the good mood with respect to a tool.

- A shift in focus to developing reusable, library-based code with task-specific modules and customizations would greatly improve the overall quality and cohesiveness of annotation systems to the benefit of the entire research community.
- Development of such an industrial strength toolkit will only be possible with sustained support for a robust linguistic annotation infrastructure that can marry the "science of annotation" with well understood principles of software engineering.
- We shouldn't imagine that we can start from scratch and impose "the one true UAT". A hybrid approach seems the most likely to succeed.
- The most important management strategy is to develop the most compelling and universal capabilities first, so that early adoption is widespread.
- To be successful, there needs to be an application for UAT from early on.

Open Source

- After a startup period, all code and other materials should be available to everybody and contributions by people outside of the core should be welcomed.
- My first inclination would be to suggest an open source development strategy. However, this doesn't always result in production level code.
- I think that an open-source code base promotes active development.

Organization

- I think it would be simpler to maintain one site that houses and chains together different modules for annotation, which could be independently maintained. If one particular annotation tool is not well maintained, then such a site would offer users other competing tools that are being maintained. That being said, it would also be important for independent tool creators to provide living guidelines on the shared site that would communicate developments and updates.
- For most approaches, it seems rather feasible to maintain peripheral functionality pluggable in some way. So the core functionality is more of a problem. None of the current systems that I know of have been developed in a way that it could be maintained by a group should the main developer no longer be available. So that certainly has to change, and definitely means quite some overhead.
- The secret to sustainability to be build a starfish rather than a spider, as in "The Starfish and the Spider: The Unstoppable Power of Leaderless Organizations"
- We should conceive of UAT as an ecology of interoperating tools, data repositories, and automated services builds on the starfish model and allows for the whole to thrive, even as the individual members wax and wane. The Open Language Archives Community offers an example in which a small amount of grant funding helped to build the infrastructure for interoperation, and now 15 years later there are 56 participating institutions each of whom are funding their own participation.
- Perhaps, by creating a working group that would oversee the development / integration of different types of tools, maintaining a common project website and code repository.

Partnerships

- I guess there must be at least two bigger funded projects, one on the US side and one on the EU side, that run in parallel and are coordinated with each other.
- Join forces with existing initiatives (e.g. LDC, Linguist List, SIL) and research infrastructures.
- Getting buy-in from different developers seems critical (which probably means leveraging some currently existing tools & formats).
- It might make sense to partner with industry on current product development like clinical coding modules (aka Nuance/IBM). Development based around common research/product goals ensures sustainability over the long run.
- If UAT is perceived as the best tool for building / managing large scale resources (like those disseminated by the Linguistic Data Consortium), then seeking partnerships with large-scale resource creators would be another strategic move to consider.

Review

- I think it is important to look at what is working now.
- We need a good survey of the current status: what is available, what are its advantages/disadvantages.

Other

- It would be great if, apart from content and engineering, we could discuss the potentials for project proposals. It may also make sense to discuss this both for the whole group and for the US and EU separately (since funding structures are quite different, I suppose).
- As I have no idea whether I am the complete outlier with my approach to annotation, I am not sure what to say. A good idea might be to start [the workshop] with what you had in mind, and then have a discussion at the end of the first day about priorities for the second.
- The workshop should accomplish the following things:
    1. Everyone in attendance agrees on what the problem is. This requires a statement of unanimous support for what our needs are.
    2. Everyone buys into one or two avenues or directions for how to solve the problem(s) mentioned in 1. This may entail some breakout sessions, to really get to what the architecture for a UAT would be, if that's what we want/need as a community.
    3. Solicit the help at the highest levels possible from members of the workshop to make the chances for success as great as possible.
    4. People should be excited by the prospects for what is being proposed.

E. Original Workshop Agenda

The following table represents the original workshop agenda, as proposed by the organizer to the participants. The actual workshop schedule differed somewhat from this schedule, as a result of dynamically adjusting the proposed schedule based on participant feedback.

Sunday, March 29, 2015		
Welcome reception on the Caracol Terrace, 8-10pm		
Monday, March 30, 2015		Workshop will be held in the **Ocean Terrace Room**
Start	*End*	
9:00 AM	9:30 AM	Introductory Remarks by Mark Finlayson
9:30 AM	10:30 AM	**Motivations: What are the problems we are trying to solve?**
10:30 AM	11:00 PM	*Coffee Break*
11:00 PM	12:30 PM	**Motivations II**
12:30 PM	2:00 PM	*Lunch in Caracol Restaurant*
2:00 PM	3:30 PM	**Capabilities: What functionalities should we target, with what priority?**
3:30 PM	4:00 PM	*Coffee Break*
4:00 PM	5:15 PM	**Capabilities II**
5:15 PM	5:30 PM	**Discussion: Should we re-work tomorrow's agenda?**
5:30 PM	6:15 PM	*Free time*
6:15 PM		*Meet in lobby to go to dinner at Timo (17624 Collins Ave.)*
Tuesday, March 31, 2015		
Start	*End*	
9:00 AM	9:30 AM	Overview of European Projects & Funding by Erhard Hinrichs
9:30 AM	10:30 AM	**Form: What form should a solution take?**
10:30 AM	11:00 PM	*Coffee Break*
11:00 PM	12:30 PM	**Incentives: How do we ensure adoption and long-term viability?**
12:30 PM	2:00 PM	*Lunch in Caracol Restaurant*
2:00 PM	3:30 PM	**Management: How should the solution be funded, built, and maintained?**
3:30 PM	4:00 PM	*Coffee Break*
4:00 PM	5:30 PM	**Discussion: Summarizing the recommendations**
5:30 PM	6:15 PM	*Free time*
6:15 PM		*Meet in lobby to go to dinner at Tony Roma's (18050 Collins Ave)*

Table 5: Original workshop agenda