



Embedding Knowledge Elicitation and MT Systems within a Single Architecture

MARJORIE McSHANE¹, SERGEI NIRENBURG¹, JAMES COWIE² and RON ZACHARSKI²

¹*Computer Science and Electrical Engineering, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA, E-mail: {marge,sergei}@umbc.edu;* ²*Computing Research Lab, New Mexico State University, Las Cruces, NM 88003-8001, USA E-mail: {jcowie,raz}@crl.nmsu.edu*

Abstract. This paper describes Expedition, an environment designed to facilitate the quick ramp-up of MT systems from practically any alphabetic language (L) into English. The central component of Expedition is a knowledge-elicitation system that guides a linguistically naive bilingual speaker through the process of describing L in terms of its ecological, morphological, grammatical, lexical, and transfer information. Expedition also includes a module for converting the elicited information into the format expected by the underlying MT system and an MT engine that relies on both the elicited knowledge and resident knowledge about English. The Expedition environment is integrated using a configuration and control system. Expedition represents an innovative approach to answering the need for rapid-configuration MT by preparing an MT system in which the only missing link is information about L , which is elicited in a structured fashion such that it can be directly exploited by the system. In this paper we report on the current state of Expedition with an emphasis on the knowledge elicitation system.

Key words: knowledge acquisition, knowledge elicitation, rapid ramp-up

1. Introduction

The development and deployment of natural language processing (NLP) systems have been significantly hampered over the years by the difficulties associated with acquiring knowledge about language for them. These difficulties can be summarized as the inability to attain good quality and coverage at a reasonable cost. Indeed, the design and acquisition of adequate-coverage lexicons, grammars and other rule sets of sufficient explanatory and discriminatory power to facilitate automatic text processing is a daunting task. A number of avenues present themselves for meeting this challenge.

First, it is intriguing to consider whether this knowledge acquisition (KA) task can, in fact, be bypassed and/or substituted with a simpler, less expensive one. Corpus-based NLP has been investigating this possibility by relying on measures of difference (distance) between texts to guide system decisions. For example, in Machine Translation (MT) between L_1 and L_2 , the distances are calculated

between elements of L_1 input and strings in the L_1 (also referred to as the source language, or SL) side of an L_1 - L_2 aligned corpus (L_2 is also known as the target language, or TL). Second, higher levels of automation should make the KA process less expensive. A variety of machine learning techniques are being investigated in the field with this purpose in mind. Third, to enhance the quality of KA today, the general level of training of knowledge acquirers should be raised. The emergence of graduate programs in language technologies is evidence that this option is taken seriously. Alternatively, computational systems can be developed that allow less well-trained personnel to carry out KA. So far, user interface technology – and generally human-computer interaction – has concentrated on ergonomic and general cognitive issues. The next step is to integrate the interfaces with metaknowledge about the knowledge to be acquired and a methodology of guiding the user through the acquisition process. This inaugurates a new knowledge acquisition paradigm that we call *knowledge elicitation* (KE).¹

In this paper we describe Expedition, a system that uses the KE paradigm to facilitate the ramping-up of MT systems from practically any alphabetic language into English in finite time.² Expedition is an interactive, web-based system that contains all the linguistic and processing information required for an MT system except for knowledge about the source language, L , which is gathered using a KE system. Expedition essentially follows classical transfer-based MT architectures. It supports sentence- and phrasal-level syntactic transfer, lexical transfer, and feature transfer. Transfer-based MT was chosen because the goal of the system was to produce “*Systran* levels” of performance in a short development time. Transfer was felt to provide the correct level of performance using well understood methods and relying on data which an unskilled informant could be expected to provide.

Expedition is comprised of four processing modules:

- the **configuration and control system** (CCS), which guides KA, manages both information resident in Expedition and information elicited during KE, tracks task prerequisites, and maintains communication among the modules;
- the **knowledge elicitation system** (Boas), which guides a language informant through the process of describing L ; it relies on resident metaknowledge about generic parameters that can be used to describe languages, their value sets, and means of realizing the latter;
- the **MT Builder**, which configures any of the three levels of MT in the system by converting the data files into the correct format for use by the MT system. This involves mapping from the internal file formats used by Expedition, which are intended for display and update, to the typed feature structure format used by the translation system.
- the **MT system**, which takes as input text in L and outputs an English translation.

One can picture Expedition as a superstructure with two embedded contentful modules – the KE system and the MT system, and two embedded support modules – the CCS and the MT Builder. Although all four modules were developed

synchronously within Expedition, they can be extracted and combined with other programs to fulfil other goals. For example, whereas the MT system can be run in an embedded fashion from within Boas in order to guide further acquisition of linguistic knowledge, it can also be run decoupled from Boas as a stand-alone application, or it can be embedded in some other toolset, such as a cross-language retrieval system.

Figure 1 illustrates the organization of Expedition. The Boas KE module relies on [1] resident knowledge about English and any online materials about *L* that can be directly imported [2] using corpus processing and dictionary format conversion tools (not shown). Boas elicits from the language informant knowledge about *L* that is passed on [4] to the MT-system builder. A command to build an MT system can be issued [5] at practically any time in the acquisition process, as the system can be built to function with incomplete open-class lexical coverage and at three different levels. Level-0 MT performs uninformed look-up of the input text words in the *L*-English lexicon and substitutes the English citation forms for those that are found. Level-1 MT is a full word-substitution system – it analyzes the input morphologically, looks up citation forms of words in *L*, translates them into English citation forms and then generates the appropriate English word forms based on a morphological feature transfer module. Level-2 adds syntactic analysis, transfer and generation. Ecological processing benefits all the MT levels.

The suggested, though not obligatory, approach to building an MT system using the Expedition environment is to provide some *L* information, ramp up the MT system, evaluate the coverage, add more information, reconfigure the system, and

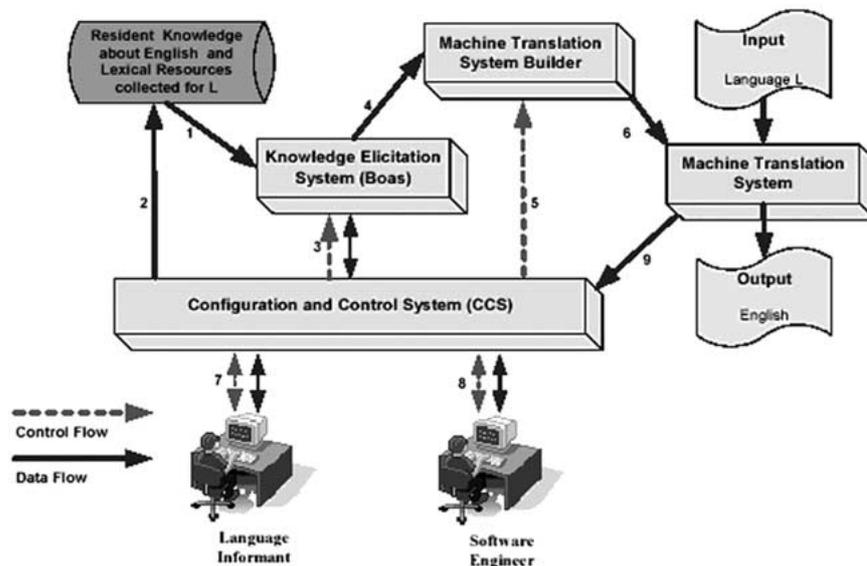


Figure 1. The top-level architecture of Expedition.

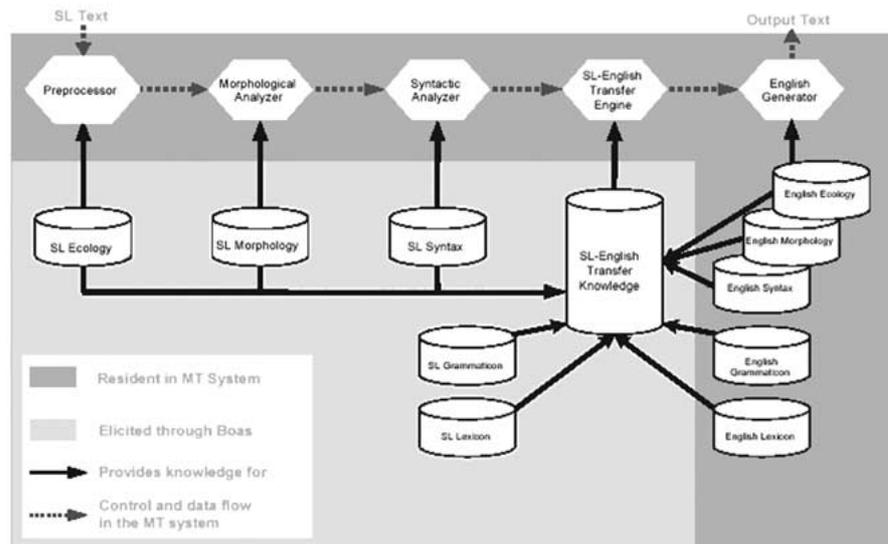


Figure 2. MT resource architecture of Expedition.

so on, until the desired level of accuracy/coverage is reached. Once the command to build an MT system is received, Expedition creates processing rules for the MT system and passes them on to the latter. At that point, the MT system can be run independently. The results it produces can be sent back to be tested [9], which can trigger modifications in the system's knowledge and another round of MT-system building. The overall process of acquiring the knowledge and building the MT system is mediated by the CCS that handles all interaction between the acquisition team and the system [2, 3, 5, 7, 8]. CCS organizes the overall operation of Expedition, tracks task prerequisites and maintains communications among the modules.

Figure 2 shows the kinds of knowledge and processors that Expedition supports. The darker shaded region contains resources resident in Expedition upon delivery to the language informant, whereas the lighter shaded region contains those that are elicited through Boas. The configuration of the processing modules is for Level-2 translation, when information about the morphology and syntax of *L* has been provided; configurations for Level-0 and Level-1 will omit and/or simplify some steps in the process.

In more detail, the processing modules illustrated in Figure 2 carry out:

- preprocessing of the source text, which involves tokenization, recognition of abbreviations, proper names, dates and numbers, and sentence segmentation;
- morphological analysis, which splits each token into a lexical stem plus features based on information elicited about the inflectional morphology (paradigmatic and agglutinating), derivational morphology, closed-class affixes and syntactic affixes of *L*;

- dictionary look-up, which includes the recognition of words, phrasal expressions, multi-word inflectional units, abbreviations and proper nouns;
- syntactic processing, whose main goals in the current implementation are to isolate noun phrases, assign them a grammatical function, and detect a limited inventory of other syntactic properties;
- feature and structure transfer, the mappings for which derive from the operation of Boas;
- English generation, which takes the feature structure produced by the analyzers and fires resident English generation rules to produce English text.

There is a complex correspondence between the processing modules and the SL and English knowledge that support them. For example, a given SL token can be a proper name (recognized as such using ecological information) that inflects (detected by inflectional morphological knowledge) and that has been listed with its English equivalent in the open-class lexicon. Thus, although information about different linguistic aspects of SL is elicited in different modules (described in Sections 3.3–3.7 for ecology, morphology, closed-class lexicon (or “grammaticon”), open-class lexicon and syntax, respectively), that information is automatically turned into an integrated inventory of analysis and transfer rules that are exploited as needed at runtime.

The functions, developers, users, results and evaluation metrics for each of the four components of Expedition are summarized in Table I.

All of the modules of Expedition maintain a strict separation between data (rules or lexical entries) and processing engines. Therefore, the processing engines can be easily adapted to new data (i.e., new source languages) and the data can be served by new processing engines.

Expedition represents a novel approach to rapid-deployment MT from the linguistic, computational, and user-modeling perspectives. Linguistic phenomena are organized in a framework of parameters, parameter value sets, and means of realizing the latter, permitting the focused, expectation-driven elicitation of information that can automatically be converted into useful NLP resources (Nirenburg, 1998; McShane and Nirenburg, forthcoming). The engines to convert the elicited knowledge into processing rules are currently designed to map to a typed feature structure representation, which is used as input to the MT system’s compilation stage. However, the intention of using the XML format as the internal representation for the description of *L* was to allow the production of a variety of shareable resources for NLP systems and for humans. The production of a descriptive primer of *L* using a standard XML parser and generator is one possibility. This would involve linking the linguistic information stored in Boas and the language information for *L*. The entire system is couched in extensive training materials, making it accessible to untrained users.

Expedition is designed to record its findings using an overt specification (at the system level and, therefore, often opaquely for the user) of abstract linguistic categories and category values. The inventory of categories and values, which de-

Table I. A concise overview of the four main modules of Expedition.

Entity	CCS (control)	Boas (KE)	MT Builder	MT system
Functions	Runs Boas, MT Builder, MT system stores linguistic knowledge	Elicits knowledge from the informant	Builds MT system	Translates <i>L</i> into English
Developers	Computer scientists, software engineers	Computer scientists, software engineers, linguists	Computer scientists, software engineers, linguists	Built by MT Builder
Users	Language informant, software engineer	Language informant, software engineer, (field linguist)	Used by full system	Information analysts examining documents in <i>L</i>
Results	See items to the right	Structured knowledge about languages	MT system	Translated texts
Evaluation	See items to the right	Breadth and depth of linguistic coverage, including language families covered	Time to build, integration of linguistic knowledge into MT components	Accuracy and fluency or translations

rives from cross-linguistic research and is resident in the system from the outset, classifies and organizes linguistic information for use by the processing modules. In strictly corpus-based approaches, neither categories nor category values are sought out; in machine-learning approaches, they are expected to be induced automatically. In the KE approach, the category (parameter) space is specified by the developers, while the actual values of the parameters for particular languages are chosen by the informant with the help of the system from the general value sets for specific parameters.

In developing Expedition, we put forward a set of ground rules, constraints and assumptions that allowed us to constrain the task to make it both feasible and useful:

- MT systems between major languages already exist. Most commonly, therefore, *L* will be a “low-density language”, that is, one for which few or no NLP resources are available and paper resources (e.g., learner’s or reference grammars and dictionaries) may also be lacking. As a result, Expedition has been designed to operate even in the absence of any linguistic resources for *L* (though the availability of such resources will help to enhance coverage and accuracy).
- The system will be worked on by *one language informant*,³ who must know *L* and English but need not have linguistic experience, and *one software engineer*, who must provide system support but need not have NLP background. The software engineer will use the KE system to obtain guidance about building tools and resources for *KA* as well as about building the actual end application, the MT system. This latter task, incidentally, is made very simple by Expedition and is expected to take minutes rather than months, as in the case of building MT systems from scratch. Relying on a small team makes the whole enterprise more feasible, as it might be difficult to set up a large acquisition team for many low-density languages. The tasks of the language informant and the software engineer are quite disjoint, so that no clashes of decisions are expected to occur.
- The MT system will have only *one TL, English*. This allows for information about transfer and generation to be resident in the system. Among other benefits, this allows us to provide adequate support for the acquisition of both open- and closed-class lexical material, as the English side of the corresponding lexicons can serve as a very good basis for KE. Acquisition of knowledge for both *SL* and *TL* would have both significantly extended the amount of time necessary for configuring an MT system and would have made the acquisition of lexicons and transfer knowledge much more complex and less reliable with respect to coverage, if simply because Expedition’s resident knowledge contains about 60,000 English open-class word senses and a complete list of closed-class word senses to help this acquisition. (Expedition incorporates tools for lexicon acquisition from corpora so that, in principle, this “priming” by English word senses is not necessary; however, if the English materials are not used, the general feasibility of the MT system being configured decreases, as the task of adequately delimiting word senses for words obtained through corpus analysis of *L* may well be beyond the capabilities of the language informant, to say nothing about the actual availability of even monolingual corpora for many of the languages that Expedition is expected to serve.)
- English has also been selected as a *common working language* in the user interface. This constraint permits some degree of English-orientation in KE and, most significantly, facilitates the preparation of a vast apparatus of online training, reference and help materials in Expedition.
- The target period of time within which each MT system should be built using Expedition has been set at about *six months*. This constraint has been imposed

to enhance utility of the system in practical applications, when a multilingual capability is required very fast. However, this time frame can be adjusted both up – leading to greater lexical and grammatical coverage by the MT system – and down – leading to some inevitable degradation in coverage and quality. The downward reduction is clearly not infinite. We expect that a meaningful system for a language with no tangible resources available will not be able to be built in less than two months. However, having more than one language informant carry out different tasks simultaneously (a mode of work explicitly provided for by the system architecture) can speed progress, especially for the time-consuming task of acquiring the open-class lexicon.

- As a corollary to the above assumption, Expedition’s design allows for the MT system to be built at practically any stage of the acquisition process. Testing the MT systems based on incomplete knowledge provides users with results (mistakes in grammatical generalizations, lexical lacunae, etc.) that can be used to steer further KA.
- The system is expected to be robust and to operate satisfactorily in a *stand-alone* fashion. No human, live or virtual, will be delivered with the system, so all training materials, elicitation functions, and processing capabilities must be resident in it.
- The importation of resources for *L*, should they exist, has been limited to on-line *mono-* or *bilingual lexicons and word lists*. The software engineer will be guided by the system through a format conversion procedure for the lexicons, as the latter must conform to the system-internal format to be usable. The word lists will be obtained through the application of the corpus processing tools resident in the system. As is the case with all the tools incorporated in Expedition, the system will guide the software engineer through the steps of generating word lists.
- The *quality of MT* will depend upon a combination of factors, some under the control of the developers (e.g., good coverage of language phenomena in KE, proper conversion of elicited knowledge into processing resources), some under the control of the informant (the quality and amount of information input), and some beyond anyone’s control (how well the language lends itself to description in a machine-oriented template system).
- The system will not attempt to gather every possible fact about the grammar and lexis of *L*, concentrating on only those facts that can be elicited in a reasonably generic fashion and automatically turned into resources for processing.

The current status of the modules of Expedition will be discussed throughout the paper as well as in Section 5. However, a snapshot of the current status will serve as orientation. The CCS control system is fully functioning, with outstanding desiderata limited to improving the “smartness” of task-redo capabilities, which involve tracking prerequisites and comparing stored information with newly input information. The Boas KE system is also fully functioning and covers what

we believe to be a solid inventory of linguistic phenomena found in alphabetic languages. Extension of Boas will involve covering more phenomena (especially less-common syntactic structures and difficult morphological phenomena, like reduplication and incorporation) and permitting non-alphabetic writing systems. All of the information elicited in Boas is stored in XML format but not all of it is currently used by the MT Builder. The MT Builder incorporates rules of flective morphology but not other morphological processes, like agglutinating inflectional morphology or derivational morphology; in addition, it does not incorporate data elicited in the ecology or syntactic modules. Therefore, Level-0 and Level-1 MT systems can be built at the push of a button, with Level-1 incorporating a subset of relevant information, but Level-2 remains under construction. Further development of the system is pending funding.

This paper is organized as follows. First we describe relevant work of others in order to place this effort in context (Section 2). Then we describe each of the four modules of Expedition in some detail (Section 3), followed by an end-to-end user's view of the system (Section 4). Next we present results to date, lessons learned and plans for future work (Section 5). Finally, we discuss further implications of this R&D effort (Section 6).

The overall methodological issues that this paper highlights include:

- methods of eliciting language knowledge for automatic conversion into processing rules;
- the use of universal and non-universal parameters and values for language description;
- design decisions for KE deriving from its application – here, MT with English as the TL;
- the division of responsibility between the system and the informant in mixed-initiative KE;
- strategies for guiding linguistically inexperienced users through the process of KE;
- practicality, or the bounds of reasonable expectations, in quick KA for quick ramp-up MT.

2. Relevant Work of Others

Boas is used to extract knowledge about *L* from an informant with no knowledge engineer present. In this, it differs from typical expert systems that rely on a personal interview with a domain expert carried out by a knowledge engineer (see, for example, Gaines and Shaw, 1993; Motta et al., n.d.). As concerns automated KE systems, most – like AQUINAS (Boose and Bradshaw, 1987) and MOLE (Eshelman et al., 1987) – are workbenches that help experts in any domain to decompose problems, delineate differences between possible causes and solutions, etc. Like typical knowledge engineers, such systems have no domain knowledge and therefore focus on general problem-solving methodologies. Other systems

permit editing of an already existing knowledge base, with the design of the editor following from a domain model. For example, OPAL (Musen et al., 1987) provides graphic forms for cancer treatment plans, which reflect how domain experts envision such plans, and these plans can be tailored by users. Boas more closely resembles the second model in that it relies heavily on a domain model; however, like the first model, it must also support not entirely predictable types of problem solving, such as analyzing language data. An important aspect of Boas is that the task set to users is cognitively more complex than the tasks attempted by many KE systems. For example, the system discussed in Blythe et al. (2001) has a user provide information about travel plans. While the challenges confronting the developers of such a system are formidable (e.g., determining whether it will be less expensive for the person to rent a car or use taxis), the cognitive load on the user is minimal. In Boas, by contrast, the user plays the role of linguist which, even under close system guidance, requires natural analytical ability and much concentrated work.

Limiting the analytical challenges posed to language informants is a goal of the KE system called AVENUE (Probst and Levin, 2002). In AVENUE, language informants are asked to translate a large inventory of sentences (currently 850, expected to grow to 10,000) then align the elements in the source and target variants; machine learning then takes over to infer transfer rules. This approach, in contrast to the one being developed in Expedition, shifts a larger proportion of the work from the language informant to the machine-learning engines (Carbonell et al., 2002).

3. System Description: Expedition Processing Modules

3.1. THE CONFIGURATION AND CONTROL SYSTEM: CCS

The principal function of the CCS is to enable users to carry out the multitude of complex tasks required to develop an MT system, from initial set-up through massive resource acquisition to final evaluation. Each task in CCS is described using a standard representation mechanism, the "file card", which represents the status of every task throughout a user's work on the system. The closest analogy to this arrangement would be to the currently emerging network management protocol standards which allow, for example, a network manager in an office in New York to query the state of work components worldwide. CCS is able, at any time, to examine file cards to determine the status of their respective tasks and, in accordance with that information, channel the user's work in the required direction. In effect, the set of CCS file cards can be seen as an overall model of the KA and MT building processes in Expedition.

CCS is accessed through a Web browser over the Internet. It makes use of the Apache web server with Java Servlet support to deliver content to users. This content is presented as a task tree in a simple, mouse-activated interface. CCS relies on a task database that organizes this complex task (building an MT system) as

a hierarchical set of subtasks, each of which is associated with specific tools and data. This database keeps track of dependencies among tasks as well as the status of each task (whether the task is completed, started but not finished, etc.). CCS supports simultaneous collaborative work, preventing potential conflicts by restricting a given subtask to one user at a time. It allows the developers to incorporate tools and applications in a range of languages including Python, Perl, C++, and Java. All elicited knowledge is saved in UTF-8 XML files.

CCS is essentially a *process manager* that facilitates the development cycle of any application of Expedition, integrating components and data. Every Expedition component can use CCS facilities, as they are implemented in the web server. This arrangement is particularly helpful for enabling the generation of uniform graphical user interface windows, for which CCS has a special mark-up language embedded in the HTML pages and interpreted by the server. This *extended HTML* (EHTML) is intended to relieve system developers of the need to implement Java or Javascript functions for the many common operations required by the acquisition process.

The CCS file cards are kept in the “CCS catalog”. Each file card represents some activity that either the software engineer or the informant can carry out. File cards have information about prerequisites and subtasks, output and input data URLs and locations of HTML tutorial files. All the information in the file cards for a particular user/language is read when a user connects to the CCS. The “task network” is an internal data structure built from the file cards. It is used to control the sequence of activities carried out by users. The user interacts with any particular file card when its EHTML page is displayed through the server. Figure 3 shows a dynamically generated CCS navigation page for the Nouns node, a descendant of the Paradigmatic Morphology node in the Expedition task tree. It has four parent nodes, listed as Ancestor Tasks, and five subtasks, the first of which can be run at this time, as indicated by the “traffic light (which is green)” icon next to it. The “do not enter” icons marking the other four subtasks indicate that certain prerequisites (in this case, the completion of the Introduction task) must be met before those tasks can run. As the user moves from task to task, the file cards are updated with the new status of the task and the current, in-memory, task network is updated.

Improving the CCS to permit better task-redo capabilities is as much a planning challenge as an implementational one.

3.2. THE BOAS KE SYSTEM

Linguistic KA is a complex task even for seasoned MT developers, but in Expedition three additional constraints raise the bar: *L* can be any alphabetic language; the language informant is not expected to be a linguist, let alone a system developer; and all elicitation, training and processing materials have to be incorporated from the outset, with no possibility of retrofitting to cater to the particular phenomena in a given *L*. Thus, both the methodological initiative and responsibility for good coverage rest largely with the system itself.

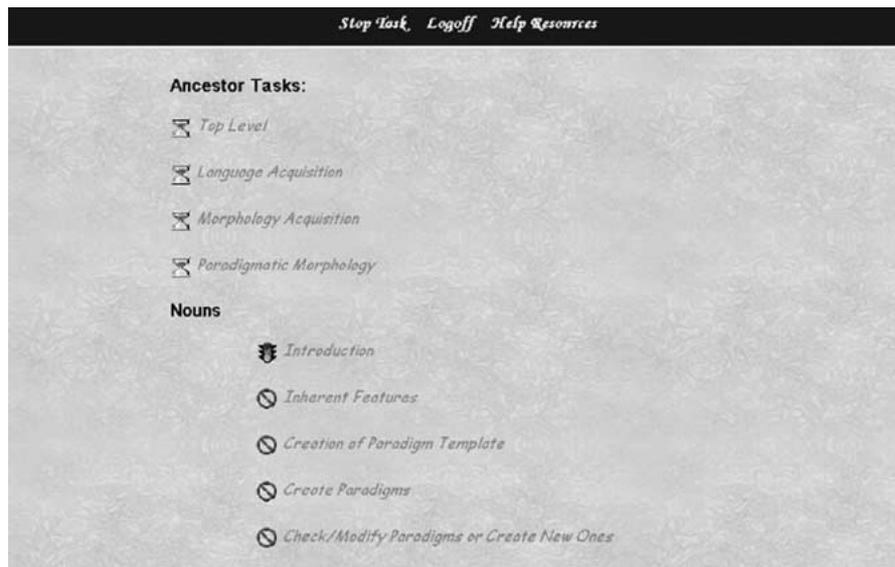


Figure 3. The view of the CCS navigation tree when the user is about to begin work on the paradigmatic morphology of nouns. The task “Introduction”, which is ready to run, is a prerequisite for the four tasks listed below it.

It is easy to perceive a similarity between the task of the Boas system and the work of a field linguist. Both in KA for an MT system and in field linguistics there is a special methodology, an inventory of lexical and grammatical phenomena to be elicited (for field linguists, this is organized as a questionnaire of the type developed by Longacre (1964) or Comrie and Smith (1977)), and an informant. There are, however, important differences. Whereas the field linguist can describe a language using any expressive means, Boas must gather knowledge in a structured fashion; and whereas the field linguist often focuses on idiosyncratic (“linguistically interesting”) properties of a language, Boas must concentrate on the most basic, most processable phenomena. The latter is in the spirit of the goal-driven, “demand-side” (Nirenburg, 1996) approach to computational applications. As a result, the coverage of language material in Boas is often narrower than that in published grammars of particular languages (e.g., many syntactic, semantic and discourse phenomena are not elicited by Boas because they cannot be expected to be processed by the MT system); however, in some cases the coverage is broader (published grammars are notorious for listing just a few examples of specific phenomena and ending too many lists with an “etc.”). Additionally, for certain phenomena Boas adopts a descriptive grain size that is finer than is typical for published grammars aimed at human users, and for certain others, a coarser grain size, the appropriate grain size of description being determined by the requirements of the processing – in our case, the MT system. For example, even though the German noun *Zentrum* has

more than one sense, there is no need to split senses in lexical acquisition through Boas because all of them are translated as the English *center*.

Three basic methodological approaches are used in Boas, depending upon the nature of information to be acquired and the current state of system development:

- *data-driven KE*, which uses English phenomena to prompt for *L* correspondences; this method, which is conceptually easiest, is used most widely in lexical acquisition;
- *expectation-driven KE*, which prompts the user to provide information about *L* based on an inventory of universal and non-universal parameters and values;
- *failure-driven KE*, which permits the supplementation of acquired knowledge in specific ways based on failures in trial runs of the MT system.

Of these methodologies, the expectation-driven one deserves the most theoretical attention since it subsumes all of the most challenging aspects of describing an unknown language. Expectation-driven elicitation in Boas is organized around a set of universal and non-universal parameters, their value sets, and possible realizations of the latter in the realms of ecology, morphology and syntax (see Nirenburg, 1998). The notion of parameters and values must be understood in a specifically NLP-related manner, not in the manner of theoretical linguistics (e.g., Chomsky and colleagues' "principles and parameters"), where the inventory of parameters and their values is far too limited for realistic large-scale applications.

Expedition's KE and MT systems deal differently with different types of parameters and values, which is in part influenced by the fact that English is always the TL. Some parameters/values are universal: they will be expected to be accounted for in *L* and will be transferred into English. Some non-universal parameters/values will be elicited for *L* even though they may not be transferable into English (e.g., grammatical case), the reason being that such information may be crucial for building a feature structure for *L*. Still other non-universal parameters/values will be required by English whether or not they are found in *L* (e.g., simple vs. continuous aspect for verbs); these are dealt with by a variety of methods, including default transfer rules (e.g., dual > plural) and what we call "bundling". The latter is a translation-driven method of establishing correspondences between complex sets of parameter values in *L* and English, a situation that arises most frequently for verbs, which can inflect for voice, mood, aspect, tense, number, gender, person, etc. For example, if a Russian informant translates the verb form *сдел* [s"el] as *ate*, the system creates a correspondence between the bundle of Russian parameter values that describe *сдел* (e.g., Past, Masculine, Singular) and the bundle of English parameter values that describes the word *ate*, which are resident in the English morphology-related resources from the outset. This method of deriving transfer rules circumvents potential non-compositionality of parameter-value bundles in *L* and gets to the very heart of the task: teaching the system to translate from *L* into English.

If Expedition is to achieve its goal of supporting the configuration of reasonable quality MT systems, it requires good information about *L*, which can come from

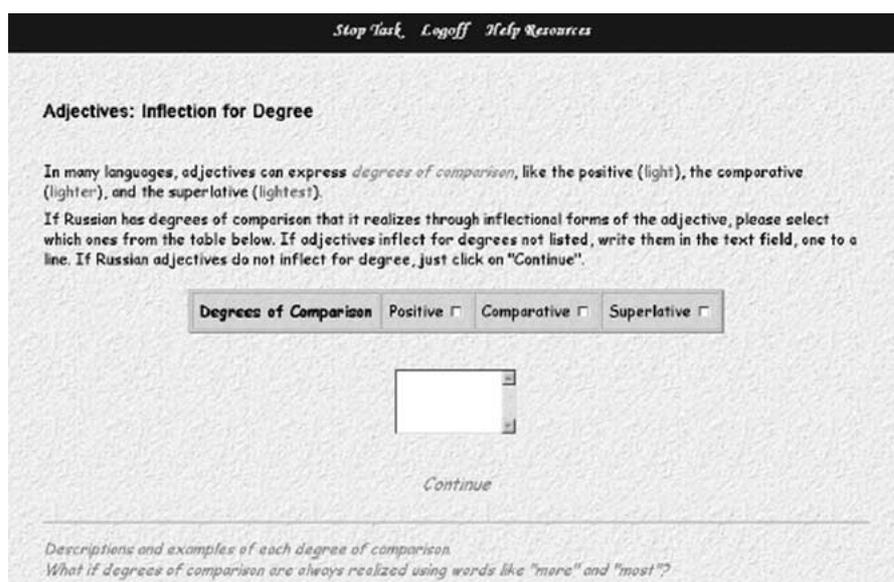


Figure 4. A sample KE page in Boas.

only one source – the language informant. Since the task is inherently complex and potentially intimidating, it is our responsibility to provide the person carrying it out with as much support as possible. Two pre-linguistic ways of doing this are greeting them with a pleasant interface and reassuring them with all manner of tutorial, help and reference materials.

There are two basic types of interface pages in Expedition: the CCS navigation pages, illustrated in Figure 3, and the pages devoted to KE and pedagogy, illustrated in Figure 4. The latter show significant variation depending on their function, but they have a single basic look and feel, enforced by style sheets that contain a number of defined styles. Each page has a tool bar at the top with three buttons: *Stop Task*, which returns users to the point in the CCS graph from which the current task began; *Logoff*, which allows them to stop the work session; and *Help Resources*, which takes them to an index of help resources, including alphabetical and hierarchical glossaries of linguistic terms, a review of icon meanings and a map of the KE system. (All of this pedagogical material, which amounts to a targeted, on-line introduction to descriptive linguistics, was developed expressly for this project and is among the resident static resources.) The remainder of the screen is devoted to elicitation tasks or explanation, with page-specific help links in the bottom left corner where warranted.

The page-level help links are just one of Boas's methods of "progressive disclosure", which permits users of various levels of experience to use the same interface. Another method is hyperlinking important terms to their respective glossary pages. For example, the italic gray reference to *degrees of comparison* in the first line of Figure 4 is a link to the glossary page shown in Figure 5.

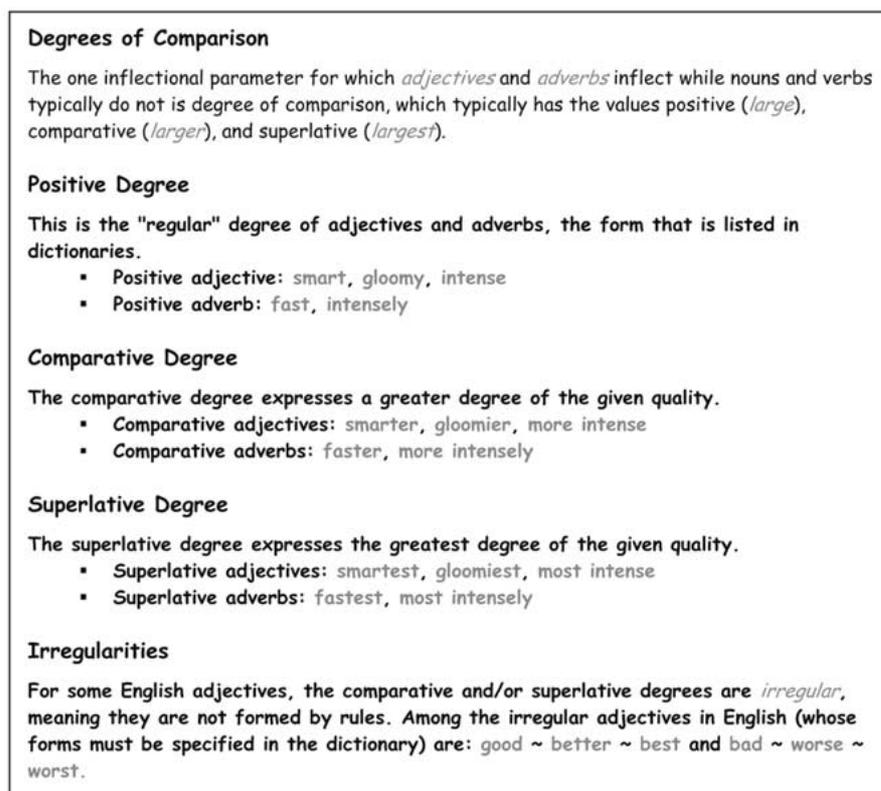


Figure 5. An example of the more than 100 glossary pages of linguistic terms in Expedition.

Glossary pages are further interlinked among themselves: for example, the page in Figure 5 links to those describing adjectives, adverbs and irregular forms.

Progressive disclosure was deemed superior to having parallel expert and novice tracks not only for practical reasons (developer time, programming complexity) but also because a good deal of *initiation* will be imposed upon all users. This was considered essential because even linguistic experts are likely not to understand completely the needs of an MT system, let alone the special requirements of Boas. However, in a number of instances expert and novice tracks are supported in Boas if a task is deemed particularly suited to the distinction.⁴

User support, in fact, begins from the very first login, when the language informant and software engineer are taken through a series of tutorials that introduce them to the system – not only its structure and inventory of functions, but also its conceptual framework and what it can and cannot be expected to produce. For example, midway through the introductory materials we describe the challenges faced by MT based on a close analysis of the sample sentence (1), whose elements are multiply ambiguous when viewed in isolation.

- (1) The Siamese cat chased the toy poodle.

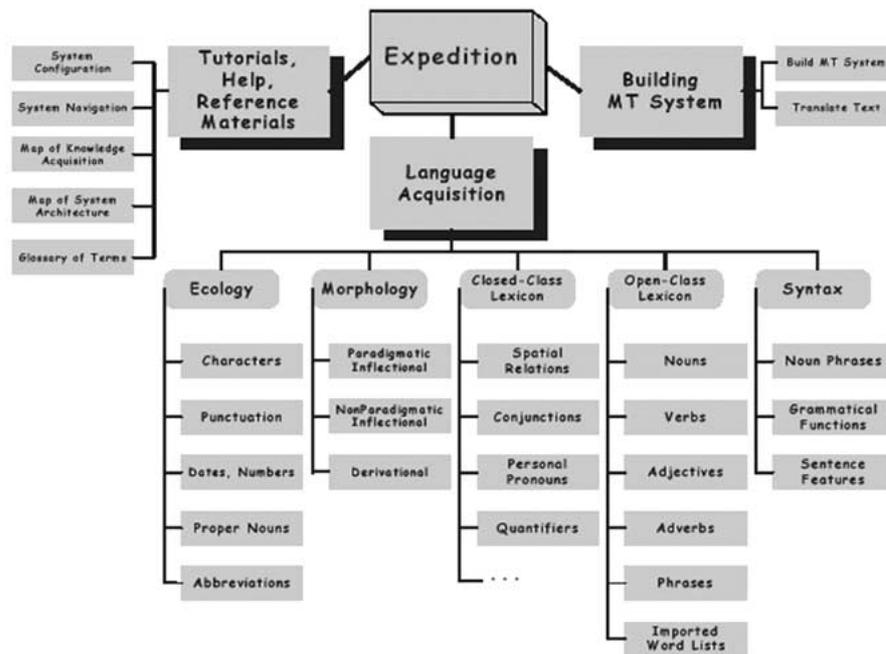


Figure 6. The top levels of the Expedition CCS task tree.

3.3. THE MODULES OF BOAS

The best way to understand how linguistic, computational processing, and MT considerations combined in the development of Boas is to survey each major module in turn. Figure 6 shows several top levels of the Expedition task tree. In this section we describe the Language Acquisition subtree.

“Ecology”⁵ refers to surface and presentation aspects of a text that lie outside the traditional realm of grammar and lexis but are crucial for effective text processing. The ecological information elicited in Boas includes the inventory of letters, numbers, and punctuation marks in *L*; transliteration conventions for proper nouns not found in the bilingual onomasticon (the lexicon of proper names); conventions for writing numbers and dates; where punctuation marks are placed and what meaning they carry; and various properties of proper nouns and abbreviations. In addition, the user is prompted to start building an onomasticon by (a) translating English lists of geographic terms, proper names, and abbreviations and (b) extending those lists using L-driven means (described below).

A sampling of potential cross-linguistic pitfalls that ecological, including onomasticon-oriented, information can help to circumvent includes the following:

- The fact that proper nouns (but not common nouns) are capitalized in Russian tells the system that the mid-sentence word *mir* [Mir] should be searched for in the onomasticon, yielding the space station Mir, rather than in the open-class lexicon, which would yield *world* or *peace*.

- Information about the representation of numbers disambiguates between *19.365* as ‘nineteen thousand three hundred sixty-five’ and ‘nineteen and three hundred sixty-five thousandths’.
- Information about the representation of dates disambiguates between *5/6/98* as ‘May 6, 1998’ and ‘June 5, 1998’.
- A list of abbreviations helps to disambiguate between periods used as full stops and periods used to mark abbreviations, for languages with this punctuation ambiguity.
- A list of words and abbreviations that commonly participate in multi-token syntactic entities helps the system to group into a single NP entities like *Maple Drive*, *Maple Dr.*, *Mt. Everest*, *Dr. Dr. Mueller* (as used in German), and *23 lbs.*⁶

The acquisition of onomasticon entries is one place where the software engineer can be of help to the language informant. From the outset of the work on a particular language in Expedition, the software engineer is instructed to collect any available lexical resources and a corpus for *L*, both of which can be supplemented and enhanced throughout the team’s work on the system.⁷ Once the informant has produced a basic inventory of combining forms (full words and abbreviations), the software engineer is instructed to search the corpus of *L* for lexical items co-occurring with those combining forms: e.g., searching for *President + X* might yield a list of dozens of presidents referred to in the corpus. The search space is restricted thanks to a special feature of the elicitation pages: when the informants list combining forms, they are asked to indicate their placement with respect to the conceptual head. For example, a Russian informant would enter the Russian abbreviation for ‘Dr.’ and indicate that it precedes the proper name with which it is used.

3.4. MORPHOLOGY

Morphology is the crossroads of many modules in Expedition. The morphological task on the source side of the MT system is to assign lexical and grammatical content to each token in the input text, a prerequisite for all later stages of processing. The challenge here is threefold: (a) organizing the KE process such that all morphological phenomena from all natural languages are covered, (b) facilitating the automatic generation of processing rules based on the collected information, and (c) making the elicitation requirements understandable to an untrained informant. Another desired aspect is that the informant’s time be used efficiently. If time were not a factor and resources were truly unlimited, full listing of at least some types of morphological forms (e.g., those generated by paradigmatic inflectional processes) along with their features would be possible as the minimum-complexity solution.

Morphological knowledge about *L* includes the inventory of grammatical morphemes and their features; the inventory of lexical morphemes and their meanings; the attachment properties of each morpheme (whether it is a prefix, suffix,

infix, or circumfix; what parts of speech it can attach to); and morphotactic rules (e.g., boundary alternations, like dropping English *e* to form creating from the citation form *create*). The following modules cooperatively cover morphological phenomena.

3.4.1. *Paradigmatic Inflectional Morphology*

In Paradigmatic Inflectional Morphology the informant establishes inflectional paradigms (of the well-known Latin type) for parts of speech whose inflectional forms are finite in number and/or are created using affixes that carry more than one bit of meaning (e.g., for English verbs, *-s* indicates three inflectional parameter values: Present Tense, 3rd Person, and Singular). Creating inflectional paradigms consists of building a conveniently laid-out paradigm template that reflects the required combinations of parameter values, then filling it with examples of all productive patterns of inflection (exceptions are listed in the lexicon).⁸

Selecting sample words for the machine-learning process can be done by the user in either of two ways – with system guidance, using the “scenic route”, or without system guidance, using the “fast lane”. The scenic route, which assumes that the user knows little or nothing about grouping words according to inflectional patterns, begins with a simple translation task. A semantically diverse sample of open-class entities is provided for the user to selectively translate and supply inherent features for, if applicable. This list of words is then fed through a paradigm-hypothesizing program that provisionally groups words into paradigms based on their final letter(s) (vowel/consonant) and, if applicable, their inherent features, since these are common diagnostics for paradigm membership in many languages. Once the system makes a preliminary categorization, the users are asked to judge whether the inflectional forms of the words in those groups are significantly alike. As they carry out this informal analysis, they manually accept or override the system’s suggestions. Once the sample paradigms are created, a morphology-learning program infers rules that are applied to the whole open-class lexicon.

Two morphology-learning strategies have been implemented, the first being more robust but requiring an expensive-to-distribute toolkit, which was the impetus for developing the second (see Oflazer et al. (2001) for a description of the first, and McShane and Nirenburg (2003) for a comparison of the two).⁹ The inventory of paradigms can be added to, edited, or amended at any time using a task called *Check/Modify Paradigms or Create New Ones*. Machine learning is then reapplied to incorporate the new data.

This method of eliciting inflectional morphology presupposes that inflected forms are synthetic, that is, consist of a single word. In many languages, (including English, e.g., *has been reading*) some word forms are analytical, that is, involve several words. Teaching the system rules of *multi-word inflection* involves first providing an inventory of auxiliaries and their inflectional forms, if applicable,

then linking them, singularly or in combination, to the correct form(s) of the main word. This process requires that the informant:

- (a) list the citation form of all auxiliaries used to form multi-word entities for the given part of speech;
- (b) indicate if they are fixed in form or inflect;
- (c) in the latter case, create a paradigm, supplied with parameter values, that includes whichever forms are used in auxiliary function;
- (d) group multi-word inflectional forms based on which auxiliaries they require (e.g., *have gone* and *has gone* would be in one group, but *will have gone* would be in another);
- (e) for each group, indicate how many auxiliaries are used, which ones they are, and in which order they typically occur with respect to each other and the head word;
- (f) indicate which forms of inflecting auxiliaries and which forms of the head word are used for the given group.

On the basis of this information, the system generates an inventory of what it expects should be valid multi-word combinations and the user checks it, correcting any possible errors. We expect that having the informant build up multi-word forms in this way will be faster than having them type out all the forms by hand.

3.4.2. *Non-Paradigmatic Inflectional Morphology*

In Non-Paradigmatic Inflectional Morphology the informant lists agglutinating affixes or independent words (typical of isolating morphology) that convey the same grammatical meanings prompted for in paradigmatic morphology. For example, the Turkish word *taşıttım*, which means ‘I made someone carry (something)’ contains a stem, *taşı* ‘carry’, plus three agglutinating affixes: *-t-* causative, *-ti-* past, and *-m* first singular. Agglutinating and isolating inflectional units are elicited together because the prompts are the same – the inventory of parameters and values mentioned above, and the method of indicating them is the same – typing one or more strings into a text field. The only difference is that for affixes the point of attachment must be indicated.

3.4.3. *Derivational Morphology*

In Derivational Morphology the informant lists affixes that derive one word from another in a formally and semantically compositional way; e.g., adding English *un-* to a scalar adjective always changes its polarity: *friendly* ~ *unfriendly*. Derivational morphology is difficult for machine processing because, both in terms of form and of meaning, simple concatenation often does not obtain. That is, adding derivational affixes to words often causes boundary and/or word-internal spelling changes; and even if the rules for such spelling changes could be listed (which is possible for some processes in some languages), the semantics of the resulting entity would often not be predictable, as derivational affixes are often ambiguous.

For example, *-er* in English is typically taken to be an affix that, when attached to a verb, *V*, produces a noun whose meaning is ‘the agent of *V-ing*’. However, this analysis certainly does not apply to the word *cooker*. Semantic non-compositionality like this is common not only for affixal word formation, but also when words are created by compounding, reduplication, and other word-formation processes. Had this process been completely compositional, the lexicons for computational applications could have been smaller, and analyzers would have been able to construct the correct meanings of words that are not actually listed in the lexicon. As this state of affairs does not obtain in most languages, Boas trains the informants to use corpus tools, failure-driven methods, and their own insights to create a large enough open-class lexicon to explicitly cover the most common words in *L* that are created by non-compositional word-formation processes. However, listing words with derivational suffixes in the lexicon is not a perfect solution since it will not guarantee that all derived words will be listed. For this reason, some derivational morphology phenomena are elicited in Boas, but only those for which there is a realistic expectation of semantic regularity.

The elicitation of derivational affixes in Boas is driven by an inventory of some 100 productive derivational affixes found in English, like *un-*, *anti-*, and *pseudo-*, many of which are frequently represented in the world’s languages. This bit of Anglocentricity is justified, we believe, in a system that serves to translate into English. Affixes like these may attach to one or many parts of speech and may or may not change the part of speech of the word to which they attach.

The English prompts are divided into the following semantic groups, all of which have numerous subclasses and members of those subclasses: negation, reverse, opposite; lesser degree; numerical relations; similarity; temporal and spatial relations; pejoratives and diminutives; etc. The distribution of variants for a given meaning (e.g., English *in-* vs. *im-* for negation) is not elicited since Expedition will only analyze, not generate, text in *L*.

Some derivational affixes are semantically empty or impoverished and function primarily to change the part of speech. Here, English prompts are used primarily for pedagogical purposes, since such processes are rather limited and idiosyncratic in English (e.g., the noun-to-verb change can be realized, among others, by any of the affixes marked here in bold: *referral*, *polishing*, *abdication*). Each possible source-target part-of-speech pair is prompted for. Affixes that change the part of speech are rare enough in some languages to suggest lexical listing as a better option; however, for truly agglutinating languages, productive treatment of such word-formation processes, despite the processing obstacles they pose, is unavoidable.

The final KE task in this section permits the free-form listing of any other semantically full affixes in *L* along with their English translations. The kinds of affixes we expect to be added here have meanings like: [added to a verb] ‘the place where that type of action typically takes place’; [added to a noun meaning a good] ‘the vendor of that good’; [added to a verb] ‘a person typically associated with that action, not necessarily as an agent’. Obviously, in order for the system to translate

such affixes, a generic translation must be supplied. We ask for translations using the variable *X*, like *the place where X typically occurs, the vendor of X, the person typically associated with X*. Translations like this will not, of course, produce smooth English; they will, however, produce something comprehensible and are a better fall-back position than no translation at all, should a derived word be missing from the bilingual lexicon.

The closed-class lexicon and syntax modules are other potential sources of affixes: e.g., the article *the* is translated by the Bulgarian suffixes *-mo* [to], *-ma* [ta], etc. (*mope* [more] ‘sea’ ~ *mopemo* [moreto] ‘the sea’) and the noun-phrase-component marker in Persian can be represented by suffixal *-e* (in the prepositional phrase *dar saxtemune* ‘in (the) building’, the suffix *-e* on *saxtemune* ‘building’ represents the noun-phrase connector known as *ezafe*).

The open-class lexicon is also linked to morphology in the sense that it permits the listing of entities that, although subject to morphological analysis, do not show sufficiently compositional semantics to be handled productively within Expedition (see Section 3.6).

3.5. CLOSED-CLASS LEXICAL ACQUISITION

Closed-class lexical acquisition in Boas asks the user to provide realizations in *L* of closed-class meanings based on English prompts. The prompts are divided into semantic classes that include: spatial relations; temporal relations; case relations; personal, reflexive, relative, interrogative, indefinite, predicative, demonstrative and possessive pronouns; conjunctions; articles; quantifiers; cardinal and ordinal numbers; and interrogative adjectives and adverbs.¹⁰ The reason for separating the closed class into a separate elicitation task are not just principled, but practical as well.

- Closed-class meanings may be realized not only as a word or phrase (like open-class meanings), but also as an affix or inflectional feature. For example, the English article *the* is translated by Bulgarian suffixes as illustrated above, and the English reciprocal *oneself* can be translated by the Russian suffix *-ся* [sja]: *мыть* [myt’] ‘to wash’ ~ *мытьсѧ* [myt’sja] ‘to wash oneself’. Feature realizations of closed-class meanings include the well-known use of the Instrumental case to indicate instrumental *with*: e.g., Polish *rewolwerem*, the Instrumental Singular of *rewolwer* ‘revolver’, can mean ‘(shoot, kill, etc.) with a revolver’.
- If closed-class items inflect, they often require different paradigm templates than those found for the open-class parts of speech. For example, the personal pronoun *I* has only singular forms, whereas the personal pronoun *we* has only plural forms.
- Inflection for closed-class items tends to be idiosyncratic so morphological learning will not be applied to them – all inflectional forms will be typed out (recall that the reason for using the morphological learner for open-class lexical items was to cut down the listing of inflectional forms when creating a lexicon;

Word	Example	Translation (Reminder of options)	Case	Paradigm
about (circa)	He was born circa 1060 and died about 1118.	около	Genitive	Add
after	We shall leave after breakfast.	после	Genitive	Add
at	At that time he was living in London.	в	Accusative	Add
before	John studied before the exam	до	Genitive	Add
		перед	Instrumental	Add

Figure 7. An excerpt from the temporal relations portion of closed-class lexical acquisition.

there is no *inherent* reason why the informant could not be asked to type out all inflectional forms of all open-class items as well).¹¹

- The realizations of closed-class items affect syntactic elicitation in Boas: e.g., if prepositions are always affixal and therefore never occur as separate words, they will not be potential free-standing elements in noun phrases. Because of this dependency, acquisition of information about closed-class lexical items must be completed before the acquisition of syntax is begun.

The closed-class acquisition interface was designed to provide for all possible *L* realizations of the English word senses. The look and feel of the interface is illustrated in Figure 7 using a portion of the temporal relations page, with Russian equivalents listed.¹²

3.6. OPEN-CLASS LEXICAL ACQUISITION

A truly sufficient open-class lexicon for translation purposes could contain hundreds of thousands of words and phrases from general as well as specialized fields. In fact, a well-known way of increasing translation quality is to list all manner of phrases, and even whole sentences, in the lexicon. However, building a bilingual lexicon of this scope cannot easily be incorporated into the task of quickly ramping up an MT system. Therefore, the driving force behind the lexical elicitation portion of Boas is to help the users to maximize their efforts – to choose not only what kinds of entries to include, but also how to go about amassing that inventory.

Before proceeding to acquisition strategies, the content of lexical entries must be mentioned. The minimal content of open-class lexical entries in Expedition, which derives from processing requirements, is a word or phrase in *L*, its part of speech, its English counterpart and, if applicable, its head (for phrasals), inherent features (e.g., gender), and/or irregular inflectional forms. Subcategorization frames and ontological links are currently not elicited since processing modules for them have not been developed.

There are three basic methods of lexical acquisition in Boas: translating English word senses from the resident list of 60,000 (which were culled from WordNet); importing word lists in *L* or English then translating them using our interface; importing a bilingual lexicon.

Since Boas is intended for languages for which few or no NLP resources are available, the method of translating word lists is expected to dominate the acquisition process. English-driven acquisition using resident word lists is one option. Our 60,000 entries are divided into manageable-sized lists first by part of speech then by frequency; and within each frequency group, words are listed alphabetically. Accordingly, the informant can choose to work on the most frequent words from each part of speech first, proceeding to less frequent words as time permits. Another acquisition option is for the informants to translate word lists that they and/or the software engineer compile. Such lists can be in *L* or English, can cover a specific subject area or be generalized, and can be gathered using Boas's corpus tools or any other means. Importation instructions are provided. Working from externally generated lists is highly recommended, at least as a supplement, for languages with widespread derivational word-formation processes like compounding and reduplication, for which Boas does not provide elicitation procedures because of the oftentimes non-compositionality of resulting forms.

Importation of a bilingual lexicon is a great time saver (if such a resource is available) and Boas provides the software engineer with instructions for preparing such resources for importation. This involves not only converting the lexicon into our XML format, but also either deleting information that will not be used by the system or collapsing it into the one field we provide for explanatory text. However, there is still no guarantee that all the minimal information required of each entry will be present. For this reason, all imported lexicons are passed through a checking program that verifies that all the expected information is accounted for. All entries that pass muster go to the Dictionary, which is the lexical resource used for the MT system. All entries that fall short are sent to what we call Purgatory, a holding place for unfinished entries.

Actually, there is more than one road to Purgatory. *Any* unfinished entry goes there, be the reason importation deficits, informant error (e.g., they forgot to indicate a noun's gender), or a systematic reflex of interface design – the latter, of course, requiring clarification. There are three open-class lexical acquisition interfaces in Boas, all variations on a theme. The interface where word lists are translated is, in a sense, primary because the informant is expected to spend the lion's share of time working there. This interface was designed for speed and convenience. In order not to hinder the central task of providing translations and inherent features, supplementary and potentially time-consuming tasks like providing irregular inflectional forms are postponed. Thus, in the basic interface one simply clicks on a checkbox if the given word has irregular inflectional forms; this automatically sends the entry to Purgatory. The Purgatory interface, a slight modification on the basic interface, provides supplementary functionality: clicking on the "paradigm"

Stop Task Logoff Help Resources				
Delete Row Copy Row Add Blank Row Merge Start Merge End				
Submit These Entries				
accident : a misfortune; especially one causing injury or death.	Russian: несчастный случай	Masculine	Inanimate	Paradigm: <input checked="" type="checkbox"/>
accident : anything that happens by chance without an apparent cause.	Russian: случайность	Feminine	Inanimate	Paradigm: <input checked="" type="checkbox"/>
account : a business or business relationship established to provide for regular services and dealings and other financial transactions.	Russian: счёт	Masculine	Inanimate	Paradigm: <input checked="" type="checkbox"/>

Figure 8. An excerpt from the open-class interface for the English-driven lexical acquisition of nouns.

button pops up a new window containing the same paradigm template that was constructed in morphological acquisition, where the irregular forms can be listed. The interface for the Dictionary permits editing of all fields but includes no further checking of completeness, so in working there the informant takes upon himself responsibility for maintaining completeness.

Figure 8 shows a sample word-list interface for a Russian system that exhibits a number of things about pre-lexicon KA:

- the informant posited two inherent features for Russian nouns: gender, with at least the values Masculine and Feminine, and animacy, with at least the value Inanimate;
- the informant has created inflectional paradigms for Russian, otherwise the “Paradigm” checkbox would not be present;
- the informant does not think that any of these translations has irregular inflectional forms, since the checkbox is not checked (one can change one’s mind, however, before submitting the entries). All of this information was imported into the lexicon interface from the morphology section of Boas.

3.7. SYNTAX

Syntax is a particularly complex aspect of machine processing and progress in the field has been slow. Therefore, in building Expedition we did not aim to create a maximal-coverage syntactic module; rather, we focused on eliciting and processing the small inventory of syntactic phenomena that we believed were at once most important for and most accessible to NLP. They represent three goals for syntactic analysis of *L*: finding the boundaries of noun phrases (NPs), understanding their grammatical functions (subject, topic, direct object, indirect object), and determining and representing select syntactic structures. The system automatically converts the elicited information into a rule set for the resident syntactic analyzer that creates a feature structure to serve as input to transfer.

Having rules to block off NPs in text is a basic starting point for syntactic analysis in most NLP systems, since this determines valid arcs/edges in the parser. In Boas, an inventory of all possible NP structures is generated automatically based on (a) the inventory of free-standing NP components in *L* (e.g., if articles are always affixes, they are excluded from syntactic consideration) and (b) the user's responses to simple binary-choice questions. That is, for each free-standing element in a NP, the user indicates its typical position with respect to the head (before/after/either) and which other free-standing NP elements can intervene. Each rule generated from this elicitation will directly produce only three-member rules, but they will be fed into a program that assumes their mutual compatibility (although this is not guaranteed in actuality) and produces a far larger rule set.

The second type of syntactic knowledge collected through Boas pertains to the grammatical function of NPs in a sentence. The user is asked to indicate the formal diagnostics for subjects, direct objects, indirect objects, and (if applicable) topics. Realization options include case-marking, the use of particles/prepositions/postpositions, and word order.

The remainder of the syntactic module seeks information about frequently encountered syntactic structures and sentence properties. Threads of elicitation currently cover basic negation, interrogatives, imperatives, coordination, subordination, copular sentences, tag questions, subject ellipsis (pro-drop), direct object ellipsis in coordinate structures, gapping, focus constructions, *yes-no* question-answering strategies, *there is* constructions, and a number of other syntactic structures. There is nothing special about this list: it simply represents a guess as to what information will at once be most helpful to, and most processable by, the MT system. We expect that broader coverage of syntactic phenomena will be necessary based on the results of broadscale testing, and we anticipate building separate threads of questions to elicit each outstanding syntactic phenomenon.

3.8. THE MT BUILDER

The MT system underlying Expedition is based on MEAT (Multilingual Environment for Advanced Translation; see, for example, Amtrup and Zajac, 2000; Amtrup et al., 2000). In MEAT, linguistic knowledge is represented using Typed Feature Structures (TFSs) (cf. Zajac, 1992). Descriptions of words, morphological feature sets, syntactic structures, and other linguistic objects are all represented in a uniform way using TFSs. The use of typing enforces the requirement that all legal linguistic structures be predefined. In this way we can check the validity of linguistic resources including dictionaries, morphological rule sets, and syntactic rules. The implementation of this formalism is efficient and supports between 3,000 and 4,500 unifications per second.

As a central data structure, MEAT uses a layered chart which is shared by all modules of the system. This layered chart (Amtrup, 1998) is an extension of the basic idea proposed by Kay (1973, 1996) and his colleagues. Tagging the edges of

Table II. Levels of MT in Expedition.

Level	Prerequisite	Action	Type of system
0	The user has acquired some open-class and closed-class lexical entries	The open-class lexicon and word realizations of closed-class items are converted to TFS representation and tokenization rules are generated	Word substitution (no morphology)
1	The user has completed morphology for at least one part of speech	Morphological rules (including those for morphologically realized closed-class items) are compiled into TFS representation	Word substitution with morphology
2	The user has completed the elicitation of syntax	Level-1 actions + PATR-style syntactic rules are converted to the form used by MEAT	Syntactic transfer

the chart with the type of structure an edge spans allows a number of a modules to access the same, shared, chart. When a module adds an edge to the chart, that edge is tagged as being generated by that module. Moreover, a module can have a limited view of the chart having access to only a specific tagged subset of edges.

Within the Expedition system, building an MT system requires converting the elicited linguistic information into the MEAT TFS representation. This involves mapping from the internal file formats used by Expedition, which are intended for display and update, to the TFS format used by the translation system. This process at the moment involves information losses, and an MT system can be envisaged which more optimally uses the resources produced by Boas. One of the optional goals of the system is to support multiple translation engines as targets for the acquisition process, for which resources will be needed to map the acquired information about L to the formats expected by the various MT systems.

Once prerequisites have been met, a user can elect to build an MT system at any time and at any of the three levels shown in Table II.

For example, a user can enter a few words in the lexicon, build the MT system, and test it. Then they can add more lexical entries and build another version of the MT system. In another scenario the user may complete the elicitation of nominal morphology and build the MT system in order to test the resulting morphological rules. They can then refine or expand the information provided about nominal morphology or move on to another part of speech. The system supports, and the developers encourage, such iterative development.

3.9. THE MT SYSTEM

The MT system takes an input text in L and outputs an English translation of L . Two levels of MT have been implemented using the results of work in Expedition. Level-0 translation uses word-for-word substitution without morphology. That is,

for every word in the input text in *L*, that word is looked up in the lexicon and if found the English equivalent is substituted in the output string. Given a simple lexicon shown in (2), the system will translate (3a) as (3b). If a word is not in the lexicon, the system will simply carry the source-text word into the target text. If more than one potential translation is found, the system currently chooses one randomly.

- (2) *casa* > *house*, *desorden* > *mess*, *es* > *is*, *un* > *a*, *esta* > *this*
- (3) a. *esta casa es un desorden*
 b. this house is a mess

Level-1 MT introduces morphological analysis of the source text carried out using a unification-based morphological engine. Rules for this engine are compiled as extended finite-state transducers. The left projections are character strings and the right projections are feature structures. This method differs from classical finite-state methods in that the output of a transition is unified with the output string rather than simply appended to it (see Zajac (1998) for more details). Once a word in the input string is analyzed, the citation form of that word along with its morphological features are used to find the correct translation in the English lexicon. (The English lexicon contains all the morphological forms of a given citation form.) For example, in (4a), the verb, *bebió* ‘drank’ is analyzed as [*beber*, Past, Singular, Third Person]. The citation form, *beber*, is translated as *drink*, resulting in the feature set [*drink*, Past, Singular, Third Person]. This set of features is then used to look up the correct translation in the English lexicon, contributing to the English translation of the entire sentence (4b). As was mentioned above, Level-2 translation adds syntactic analysis, transfer, and generation. This level has not been fully implemented in the current version of Expedition.

- (4) a. *Juan bebió cerveza.*
 b. Juan drank beer.

4. End-to-End User’s View of Expedition

The system presented to the team of informant and language engineer will consist of well-defined tasks. The language engineer will install the system based on the installation instructions. The language informant will then carry out the tasks of the Boas KE system, in effect answering an extensive series of questions, some of which require off-line preparation (e.g., delineating all productive inflectional paradigms in *L*) but most of which do not (indicating which punctuation marks can end a sentence, translating words, selecting the means by which *L* indicates subjecthood). More than one informant can simultaneously carry out KE tasks – a multi-user option provided for by the CCS that is particularly useful for the time-consuming process of building the open-class lexicon. The knowledge engineer

will assist in the KE process by importing lexicons, if available, and working with corpora, if desired. It is not expected that the team will extend the scope of the KE system, though there is nothing that prevents such innovation should they be experienced in NLP.

The only requirements of informants are that they be bilingual with English as one of the languages, possess natural analytical ability, and approach the project with a willingness to learn. We have carried out only limited testing of the system with non-linguist informants (some computer-science-oriented participants in a language engineering summer school and the programmers who worked on the system), but results of our resident pedagogy seem promising.

We do not yet have a separate module that diagnoses types of errors and automatically presents the user with means of correcting them. Instead, the user must understand which portion of the system requires supplementation (e.g., if a word form is not found, lexicon or morphological supplementation could be required). Practically all the subtasks in Boas can be supplemented at any time, after which MT Build can be rerun. The only non-trivial restriction concerns prerequisites. Say, for example, an informant creates many inflectional paradigms for nouns then at some later time realizes that he has forgotten to include some value for case. Adding that value for case, which amounts to reconfiguring the entire paradigm template, is not simple because of the sequence of prerequisites: selecting the complete inventory of parameter values is a prerequisite for building the paradigm template, which is in turn a prerequisite for providing examples of inflectional patterns.

Expedition also does not provide a means of supplying information that Boas does not specifically elicit: for example, *L* might use a syntactic construction that is not covered. The reason no such interface is provided is that we have no way of foreseeing how such information could be automatically turned into processing rules. Our plan is to create elicitation threads for all such phenomena based on user feedback, which would be a long-term project whose results would amount to patches for the original system.

5. Results to Date and Plans for Future Work

Boas has undergone continuous informal testing by the authors as well as by students and colleagues at various stages of its development. Students at the 1999 CRL Language Technologies Summer School at New Mexico State University, most of whom knew a second language natively or well, created a short profile of that language as a laboratory exercise. Students of the African Languages Center of the University of Maryland Eastern Shores used the system to develop profiles of Yoruba and Ibu, and a student at Purdue University used the system as part of a linguistically-oriented introduction to Swahili.¹³ The drawback of most of these tests is that time did not permit students to read and absorb all of the instructional materials. So, although most tasks were sufficiently understood by most users, the

work would have been easier and fewer questions would have arisen if time had permitted the system to be employed in the way it was intended – over a 6-month period of time.

The student comments, in conjunction with comments from colleagues who have viewed and tested the system, led to changes including:

- improving the look and feel of the interface;
- developing a map of the system that previews what types of information are elicited at what points in the process; this was a point of concern for many users, who would think of a phenomenon and would either want to provide information about it immediately or would fear that the system would never get to it (usually we had, in fact, planned for it);
- extending explanatory materials to target particularly difficult issues; for example, in some cases it is possible to provide the same information in more than one place, in which case the user can choose to provide it in one module, the other module, or both;
- demoting some explanatory materials to links rather than permitting them to occupy valuable screen space;
- devoting more attention to the elicitation of agglutinative morphology;
- augmenting the inventory of parameters and values;
- fundamentally redesigning the open- and closed-class interfaces to increase speed of acquisition.

It must be said, however, that the most demanding users were the developers themselves, so no revolutionary changes were made on the basis of outside input.

The results of Boas have not yet been used to ramp up full-scale MT systems but we did create a toy profile of Polish that supported throughput at translation levels 0 and 1.

To summarize, the languages that have been described to various degrees using Boas include: Bulgarian, Finnish, French, Georgian, German, Hebrew, Ibu, Japanese, Persian, Polish, Russian, Spanish, Swahili, Turkish, Ukrainian and Yoruba.

One natural avenue of development for this system would be to have English as the SL and the other language as the target. This intriguing possibility could be addressed at several levels. First, could we use Expedition, as is, to produce a translation system in the reverse direction? This might be helped by the fact that we can carry out a very rich syntactic analysis of English, possibly even to the level of distinguishing many word senses. Many details of *L* however, are not captured by the current process (for example vowel harmony or extensive syntax). Our best hope might be to produce a system that produced some kind of pidgin version of *L*. Whether, this would be sufficient to allow, for example, web browsing by a speaker of *L*, is an experimental issue that would need to be resolved. It is certainly an experiment we would be interested in trying, assuming funding were available.

The XML files that store all data generated using Boas are available and can be applied to MT or any other task. An excerpt from the XML file from the open-class

lexicon of a profile of Polish is shown in (5). Similar XML files are produced for all other types of information elicited in the system.

```
(5)
<Entry>
  <L>
    <CitationForm>drzewo</CitationForm>
    <Type>word</Type>
    <gender>masculine</gender>
    <OtherInherentFeature>
      <Name>virility</Name>
      <Value>inanimate</Value>
    </OtherInherentFeature>
    <PoS>noun</PoS>
    <Paradigm></Paradigm>
                                     ; there are no irregular forms
  </L>
  <English>
    <CitationForm>tree</CitationForm>
    <Type>word</Type>
    <PoS>noun</PoS>
  </English>
</Entry>
```

Linguistically-related lessons learned through developing Boas involve achieving a better understanding of the very nature of language description and “airing out” issues that have become stagnant. For example, although we have not discovered any hitherto unknown types of word structure, the picture we paint is quite different from existing treatments. In an environment where established schools, theories and perspectives dominate, such novelty may provide a springboard to greater descriptive coverage and a finer grain size of description.

There are many avenues of further work and enhancement for the Expedition environment. They include extending the resident metaknowledge in the system to cover additional phenomena in languages (e.g., noun incorporation or supra-segmental phenomena), developing “smarter” machine learning environments and further improving the utility and ergonomics of interfaces.

What would need to be done to extend Expedition for interlingua-based MT? Using Expedition to produce an interlingua-based MT system might be possible. The problem here is the additional knowledge that the acquirer either needs, or needs to be shielded from. The target of the acquisition process would shift to building data that would allow the MT system to translate from *L* to the interlingual representation. Two approaches are possible. In one the acquirer needs to be taught what the targets in the interlingua are, and supported in acquiring this information. In the other the Expedition system builders must still present the acquirer with

English examples and usage and use these examples to map to the metainformation of the interlingua. This would be a very interesting project, but almost certainly would be extremely prolonged and of limited success. A better approach would perhaps be to assemble a team of computational linguists, not necessarily experts in *L*, who would use the base level provided by Expedition MT and extend this manually to support MT based on a richer foundation of semantics and pragmatics.

6. Conclusions and Further Implications of this R&D Effort

The quality of MT systems directly depends on the quality (depth) and quantity (coverage) of the knowledge they use. This dependency is valid for corpus-based, statistical methods as much as for representation-oriented, rule-based methods (the former perform better when they rely on better-aligned and larger corpora). The high cost of acquiring such deep and broad knowledge can be alleviated (a) by increasing the levels of automation, (b) by using recorded metaknowledge, for example, the knowledge of universal and language-dependent linguistic parameters, to speed up the acquisition process and (c) by developing user interfaces that allow acquisition by less highly trained personnel. The Expedition project integrates all three of these methods: it involves machine learning methods, a detailed specification of linguistic parameter/value inventories for many languages and language families, and an advanced human-computer interaction methodology that we refer to as KE.

Machine learning, metaknowledge acquisition and representation and user interface technologies are vibrant research areas in their own right. However, none of them in isolation promises to alleviate the ever-present obstacles of broad coverage, high quality and low development cost for practical applications any time soon. We believe that the hybrid approach to acquisition adopted in Expedition is the best practical KA methodology today, as it is not plausible in the foreseeable future to obtain high quality KA for MT without any human participation.

It is clear that in some types of KE applications it will be difficult to develop an interface that obviates the need for the user to learn the metalanguage in which the knowledge they impart to the system is encoded. Boas did not require users to know the metalanguage (XML), since developers provided rules that generated metalanguage expressions from HTML forms filled out by the user. Some other application may require users not only to know the content of some subject domain but also to be well-versed in expressing their knowledge through the system's metalanguage.

It is not at all a trivial task for experts to be able to express their knowledge in *any* language. How many times did we hear the opinion that "I'd rather do it myself; it's too much trouble explaining things to others"? It is not only the perceived inability of people to learn that underlies this state of affairs. To use another popular simile, remember what happened to the centipede, arguably, an expert in many-legged locomotion, when somebody asked him how he manages to operate so many legs at once? So, systems that extend the capabilities of Boas must help users both

to understand how best to formulate their knowledge and, if necessary, to express it in the metalanguage used by the system.

A good example of an area where such capabilities would be beneficial is in the acquisition of ontologies, including ontologies to support NLP in specialized domains (e.g., bioterrorism, nuclear physics). This task requires domain knowledge available only to experts. But since such experts are usually not trained ontologists, recording the relevant knowledge using the expressive means available in the given ontological system is a logjam, usually necessitating the guidance of an ontologist who asks the expert the right questions in the right order.

We believe, however, that a KE system of the Boas class can be designed such that it facilitates ontology acquisition in both its content and metalanguage aspects, turning the task of the domain expert into traversing a series of well-defined questions and choices. So, whereas in the current version of Boas the parameters, values and realizations are of a linguistic nature, in ontological acquisition they could be oriented toward procedures for organizing and encoding knowledge in an ontology, supported by the same types of progressive-disclosure assistance as were developed for Boas.

We believe that Boas could be readily applied to various realms, including, for example, education. With relatively minor augmentations, Boas could support training in general linguistics, computational linguistics and field linguistics, since working through the process of providing information about a language in a structured manner would be a hands-on means of learning linguistic content and developing discovery skills. When modified for this purpose, the Boas system would:

- prepare students to work creatively and independently as linguists;
- permit a customized, user-modeled approach to problem solving;
- offer a truly empirical basis for learning;
- promote a flexible definition of “success” since the language chosen and the user’s knowledge of it would need to be taken into consideration for purposes of evaluation;
- encourage students to think globally, since rare languages will be more interesting research candidates than better studied languages; and
- facilitate the interaction between NLP and linguistics, since the content covered and means of covering it are largely driven by the ultimate processing needs.

Acknowledgements

A number of colleagues have contributed to the Expedition project over the years. We would like especially to thank Victor Raskin, Stephen Beale, Stephen Helmreich, Kemal Oflazer, Svetlana Sheremetyeva and Rémi Zajac. Many thanks also to the anonymous referees for very useful suggestions and to the editors of the special issue for the best constructive criticism and support that any of us can remember.

Notes

¹ There is no universal agreement about the meaning of the terms “knowledge acquisition” and “knowledge elicitation”. We do not attempt to compare and clarify terminological usage beyond stating that elicitation centrally involves system initiative and, therefore, relies on significant amounts of metaknowledge in the system.

² Although Expedition accepts any alphabetic character set (Cyrillic, Hebrew, extended Latin, etc.), some languages widely use grammatical phenomena that are extremely difficult to elicit and process, like noun incorporation and reduplication. While one could develop an MT system for such languages using Expedition, its quality would likely not be very good at present.

³ In this paper we variously refer to the person fulfilling this role as “(language) informant” and “user”.

⁴ For example, in the module that elicits inflectional paradigms there are both system-guided and user-independent methods of creating the inventory of sample words, as described below.

⁵ “Ecology” is Don Walker’s term relating to issues connected with writing systems, text mark-up, punctuation, special symbols, dates, numbers, proper names, etc.

⁶ Prompts for abbreviations are divided into the following rough-grained semantic classes (examples are drawn from much longer lists): days/months; times, weights, measures (*a.m., lb., m., P.O. Box*); books, writing, correspondence, media (*pg., P.S., ASAP, ch.*); locations (*St., Apt., N*); titles (*Dr., Jr., Mrs., Pres.*); companies and organizations (*Ltd., & Co., Dept., Univ. of*); miscellaneous (*KGB, b. ‘born’, DOA ‘dead on arrival’*). Prompts for proper names are world leaders. Prompts for geographic locations are continents, countries, capitals, bodies of water, etc.

⁷ A set of corpus processing, data format, language codeset conversion, and web spidering tools is configured with the current implementation of Expedition. This set will be enhanced in future implementations.

⁸ See McShane (forthcoming) and McShane and Nirenburg (2003) for details of the paradigm-building process.

⁹ The morphological-learning program works on single-word entities; multi-word entities are elicited in a separate module, where they are built up as a combination of {Auxiliary X in forms A, B, C...} + {head word in forms J, K, L...}.

¹⁰ A given sense is elicited separately when used as different parts of speech in order to avoid presenting syntactic apples and oranges simultaneously to the informant (e.g., *before* functions as a conjunction in *He turned the lights out before leaving* and as a preposition in *before November 17*), and in the hope that the informant will not forget about one or more of the uses of the given sense.

¹¹ In some languages some closed-class items inflect according to productive rules. A useful acquisition option would have been to permit the user to exploit machine learning even in the closed class, and this enhancement can be incorporated in future implementations of the system.

¹² The two Russian realizations of English *before* represent near synonyms that, by virtue of so-called “quirky” lexical specification, require that their complements have different types of case-marking.

¹³ The student is Katrina Triezenberg, working under Victor Raskin at Purdue University.

References

- Amtrup, Jan W.: 1998, *Maschinelles Dolmetschen mit Mehr-Ebenen-Charts* [Machine interpreting with multi-layered charts], Ph.D. thesis, Universität Hamburg.
- Amtrup, Jan W., Karine Megerdoomian, and Rémi Zajac: 2000, ‘Rapid Development of Translation Tools: Application to Persian and Turkish’, in *Proceedings of the 18th International Conference on Computational Linguistics COLING 2000 in Europe*, Saarbrücken, Germany, pp. 982–986.

- Amtrup, Jan W., and Rémi Zajac: 2000, 'A Modular Toolkit for Machine Translation Based on Layered Charts', in *COLING-2000 Workshop: Using Toolsets and Architectures to Build NLP Systems*, Luxembourg.
- Blythe, J., J. Kim, S. Ramachandran, and Y. Gil: 2001, 'An Integrated Environment for Knowledge Acquisition', in *International Conference on Intelligent User Interfaces*, Santa Fe, New Mexico, pp. 13–20.
- Boose, J. H. and J. M. Bradshaw: 1987, 'Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge Acquisition Workbench for Knowledge-Based Systems', *International Journal of Man-Machine Studies* **26**, 3–28.
- Carbonell, Jaime, Katharina Probst, Erik Peterson, Christian Monson, Alon Lavie, Ralf Brown, and L. Levin: 2002, 'Automatic Rule Learning for Resource-Limited MT', in S. D. Richardson (ed.), *Machine Translation: From Research to Real Users: 5th Conference of the Association for Machine Translation in the Americas, AMTA 2002*, Springer, Berlin, pp. 1–10.
- Comrie, Bernard and Norval Smith: 1977, 'Lingua Descriptive Studies: Questionnaire', *Lingua* **42**, 1–72.
- Eshelman, L., D. Ehret, J. McDermott, and M. Tan: 1987, 'MOLE: A Tenacious Knowledge Acquisition Tool', *International Journal of Man-Machine Studies* **26**, 41–54.
- Gaines, B. R. and M. L. G. Shaw: 1993, 'Eliciting Knowledge and Transferring it Effectively to a Knowledge-Based System', *IEEE Transactions on Knowledge and Data Engineering* **5**, 4–14.
- Kay, Martin: 1973, 'The MIND System', in R. Rustin (ed.), *Natural Language Processing*, Algorithmic Press, New York, pp. 155–188.
- Kay, Martin: 1996, 'Chart Generation', in *34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, CA, pp. 200–204.
- Longacre, R. E.: 1964, *Grammar Discovery Procedures*, Mouton, The Hague.
- McShane, M.: forthcoming, 'Redefining "Paradigm" for Computer-Aided Language Instruction', to appear in *Foreign Language Annals*.
- McShane, M. and S. Nirenburg: 2003, 'Blasting Open a Choice Space: Learning Inflectional Morphology for NLP', *Computational Intelligence* **19**, 111–135.
- McShane, M. and S. Nirenburg: forthcoming, 'Parameterizing, Eliciting and Processing Text Elements Across Languages', forthcoming in *Machine Translation*.
- Motta, Enrico, Tim Rajan, and Marc Eisenstadt: n.d., 'A Methodology and Tool for Knowledge Acquisition', Technical report, Human Condition Research Laboratory, Open University, Milton Keynes, UK. Available at citeseer.nj.nec.com/159278.html.
- Musen, M. A., L. M. Fagan, D. M. Combs, and E. H. Shortliffe: 1987, 'Use of a Domain Model to Drive an Interactive Knowledge Editing Tool', *International Journal of Man-Machine Studies* **26**, 105–121.
- Nirenburg, Sergei: 1996, 'On Supply-Side Vs. Demand-Side Lexical Semantics', in *Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz, CA.
- Nirenburg, S.: 1998, 'Project Boas: "A Linguist in the Box" as a Multi-Purpose Language Resource Paper', in *Proceedings of the First International Conference on Language Resources and Evaluation*, Granada, Spain, pp. 739–745.
- Oflazer, Kemal, Sergei Nirenburg, and Marjorie McShane: 2001, 'Bootstrapping Morphological Analyzers by Combining Human Elicitation and Machine Learning', *Computational Linguistics* **27**, 59–85.
- Probst, Katharina and Lori Levin: 2002, 'Challenges in Automated Elicitation of a Controlled Bilingual Corpus', in *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-2002)*, Keihanna, Japan, pp. 157–167.
- Zajac, Rémi: 1992, 'Inheritance and Constraint-Based Grammar Formalisms', *Computational Linguistics* **18**, 159–182.

Zajac, Rémi: 1998, 'Feature Structures, Unification and Finite-State Transducers', in *FSMNL'98, International Workshop on Finite State Methods in Natural Language Processing*, Ankara, Turkey.

