

Improving Communication in E-democracy Using Natural Language Processing

Michele Carenini, *University of Edinburgh*

Angus Whyte, *Napier University*

Lorenzo Bertorello, *City of Bologna*

Massimo Vanocchi, *YANA Research*

The E-democracy European Network project applied natural language processing to improve communication between public administrations and their citizens.

E-democracy—the design and development of new techniques for improving communication between public administrations (PAs) and citizens—is a major application field for natural language processing and language engineering. Helping citizens access information in a friendly, intuitive way (that is, using their own language) is the

primary objective of a global e-democracy framework. If we take NLP and language engineering out of laboratory research and into a real communication context, e-democracy can represent the ultimate testbed for different tools and techniques—no prefabricated examples to analyze, no “microworld” scheme to adopt, just real interaction with real users, on real repositories of information.

NLP aims to develop models of how humans comprehend and produce natural language to build sound theories of the relationship between the mind and language (that is, theoretical NLP) and to improve and facilitate human-computer interaction (that is, applied NLP). NLP can then be a powerful instrument to access structured and unstructured information and to improve interaction between users via computers.

The E-democracy European Network project (EDEN; see the sidebar) aimed at discovering whether a particular NLP approach could further e-democracy by increasing citizens’ participation in the decision-making process. Our goal was twofold: to test whether we could meet e-democracy requirements using

advanced linguistic technologies and to test whether Augmented Phrase Structure Grammars (APSGs) were robust and well-assessed enough to use in a real-world (and highly sensitive) environment. Thus, we tried to answer one basic question—whether and how our chosen NLP approach could improve communication in an e-democracy context. We also aimed at developing two toolsets to improve communication between PAs and citizens in the context of urban planning: a set of NLP-based tools to simplify access to information and knowledge and a set of forum and polling devices.

Improving communication in e-democracy

The EDEN research team planned to develop software modules that were based on NLP technologies and structured according to the users’ (that is, the PAs’) different tasks. We conceived the set of modules as a toolset comprising five tools:

- *Address Guesser*, which automatically selects the

proper PA office to deliver a message to on the basis of the message content;

- *Answer Tree*, which manages FAQ lists and retrieves the questions and answers most similar to the user's question;
- *Style Enhancer*, which analyzes a document's style to improve its readability before the PA publishes it on its Web site;
- *Multilanguage Helper*, which links technical terms to their translations in different languages, providing simple definitions; and
- *Natural-language Map*, which retrieves texts and maps through a natural-language interface.

We could consider Style Enhancer and Multilanguage Helper to be a single module where the PA officer writes a document and passes it to the Style Enhancer, which highlights technical words and phrases and sentences with overly complex syntactic structures and then suggests a simplified alternative. The officer makes corrections using the suggestions. If the officer decides to leave technical terms in the document, they will be accompanied by both a translation and a definition.

The project developed the toolset over three years. During the pilot phase, the project deployed the tools on the PAs' Web sites, which let citizens test them on a trial basis and also let internal users (the PA officers) use the tools to respond to citizens. We integrated the toolset in several ways to constitute a single complex system at each PA's disposal. So, for example, suppose a citizen sends a message to a PA. The Answer Tree can identify the question and may find it similar to one already answered in its FAQ list. In this case, the system returns the answer from the FAQ list while the user waits for a direct answer from the PA. If there isn't a similar question in an FAQ list, the question is routed to the proper office.

Urban planning professionals associated with one PA and an internal editorial team tested the Style Enhancer. We developed and tested the toolset progressively, ending with the Natural-language Map and Multilanguage Helper. However, we didn't include these in the pilots because of delays that we discuss later.

We designed the tools to adapt to other contexts than PAs and other domains than urban planning. Indeed, five months after EDEN ended, the EU eTen program approved a new project called HANDS (Helping Answers Decision Services), which aims at deploying some of EDEN's NLP tools.

To translate the real-world complexities of

The E-democracy European Network Project

EDEN began in February 2001 and ended in January 2004. The consortium included 11 partners: the City of Bologna, Omega Generation, YANA Research, Archivio Osvaldo Piacentini, Napier University's International Teledemocracy Centre, Nisko Info Centrum,

Digipolis, Vienna City Administration, Freie Hansestadt Bremen, the University of Bremen, and Public Voice Lab. These partners represented six European countries: the United Kingdom, Italy, Germany, Austria, Belgium, and Poland.

citizen-PA communication into an appropriate form for Unified Modeling Language (UML) modeling and then develop the toolset's functional specifications, we adopted a quick and easy requirements approach. The complex set of users included four PAs dealing with four languages and end users ranging from PA employees to individual citizens. We had to broadly define the application areas at the project's start and then quickly document the requirements and turn them into functional specifications.

We used a methodology from Napier University's International Teledemocracy Centre (our academic partner for defining methods of collecting and analyzing user needs). The methodology was in turn influenced by Soft Systems Methodology and scenario-based design. Drawing on these, the PA project members created questionnaires and interviews with situation-focused scenarios or usage narratives. They gathered comments on the envisaged NLP toolset's desired functionality from colleagues responsible for citizen communications. The technical partners documented user requirements as high-level statements of each toolset's desired functionality, as a description of the technical skills necessary to use the EDEN tools, and as a description of the corresponding business-process changes the PAs needed to successfully deploy the e-democracy tools. The technical partners then used this requirements statement to define use-case scenarios for the functional specifications.

We performed cyclic testing phases to determine whether the developed tools were acceptable to the pilot sites' project members. Early in this process, however, while we were developing the software's Alpha version, the PAs' priorities changed. Just as the tools were coming closer to reality, the PA partners—having reflected on changes in their software environment and communication needs during the requirements phase—came to regard the toolset as a set of distinct tools. Some of

these tools remained high-priority needs, but others faced internal competition from non-NLP technology. This complicated and delayed the mapping of high-level requirements to functional requirements because each pilot site now required its own combination of tools, and each combination would in turn need to operate in software environments that were also rapidly changing.

For clarity, the partners agreed to revisit the user requirements they documented in the project's first phase and to produce an elaborated (and hopefully definitive) version. The point was to reformulate the user requirements and map them again to functional specifications. So, the project had to convincingly—and quickly—turn a previously technology-driven approach into a requirements-driven approach. The technical partners adopted a structure modeled on IEEE Standard 830-1998 to unambiguously define what the PA partners needed each tool to do in terms of functionality, operational characteristics, and usability.

Articulating these updated requirements through expert panels convened by each city partner and apportioning them to the functional specifications was the project's most delicate and crucial phase and its concrete turning point. Complicating the process was the need to consider resource availability and time constraints. Ultimately, the PAs considered the completed project successful because it

- formalized the user requirements mapping to functional specifications;
- maintained an efficient communication flow and created dedicated tools to ease and support communication (for example, an EDEN glossary);
- created a clear and formal project structure in terms of partners' roles and responsibilities, which facilitated conflict and negotiation management;
- adopted the toolset that emerged from those negotiations; and
- enabled the partners to translate the changed

user requirements to real NLP applications and thoroughly test them to assess their concrete market prospects.

Technological highlights

EDEN was ultimately effective at developing a mutual understanding between technicians and users about their respective fields of interest and expertise. (From this point on, we refer to the participating PA officers as “users” and to citizens as “end users.”) This mutual understanding was one of EDEN’s main achievements.

Fast prototyping

The fact that we used the same output format without major problems in developing the Dutch, English, Italian, and German parsers was the project’s first interesting result. More remarkably, development costs were extremely low. The most significant example was how Stefan Rijnhart of Digipolis developed the Dutch grammar in just a few months. Two major factors made this possible:

- the availability of an advanced dedicated tool for grammar development; and
- the simplicity of the linguistic-processing approach, which didn’t require complex rules, complex output structure, full sentence recognition, or specific lexicon format or content requirements.

This experience is much more significant when considering the underlying linguistic approach. Rule-based analyzer development is usually a resource-consuming activity in linguistic engineering. The EDEN project demonstrated that combining the right approach with enhanced tools lets you quickly develop full-functioning parsing applications. Another interesting outcome concerns the output format. Linguistic analyzers embedded in EDEN NLP tools represent sentences by means of flat (that is, no structure, hierarchy, or explicit internal link) lists of triples. A *triple* represents a grammar relationship between two main phrases (which are generally nominal, verbal, or adjectival) as well as between any kind of embedded object. A triple is defined as

< Gr_type governing_head governed_head >

where

- Gr_type is a label for grammatical relation (such as subject, direct object, or relative clause; the grammar developer arbitrarily

defines the list of labels),

- governing_head is the main word in a phrase or clause, and
- governed_head is the main word in a phrase or clause syntactically embedded in the represented phrase.

For instance, the sentence “The dog barks” will generate the following triple:

< subject BARK DOG >

while the sentence “The man who is coming is a teacher” will generate the following triples:

< subject COME MAN >
< noun-modifier MAN TEACHER >

EDEN was ultimately effective at developing a mutual understanding between technicians and users about their respective fields of interest and expertise.

This output format clearly borrows from the EAGLES (Expert Advisory Group on Language Engineering Standards) initiative’s proposed modeling for linguistic resources standards and was first tested in the SPARKLE (Shallow Parsing and Knowledge Extraction for Language Engineering) project; nonetheless, it’s surprisingly effective and up-to-date. Its simplicity and flexibility make it independent of domain and language and easily customizable; at the same time, it can appropriately encode all information necessary for W3C RDF specifications, which are the leading Web sites and standards to the next generation.

Grammar reusability

In a 2002 EDEN report, John Bateman noted,

Re-use is generally made easier when the linguistic accounts are stratified into relatively autonomous levels of grammar, semantics and interpretation and, further, when within the grammar there is additional explicit separation

of syntagmatic and paradigmatic information. This is approximated in grammar frameworks that make a clear separation between constituent structure and feature descriptions.¹

The project’s specific grammar format separates these two information levels. Each grammar rule has both a syntagmatic part that corresponds to a reduction rule and also a set of expressions (or *actions*) that independently build each syntactic phrasal constituent’s feature structure and output structure. The grammars’ internal layout positively affects grammar reuse in two ways:

- we’ve embedded the same linguistic analyzer in several different tools, and
- we’ve conducted a grammar reuse experiment (from Dutch to German) successfully.

We can roughly divide the EDEN NLP tools into two groups:

- information access (indexing and classification technology), which includes Answer Tree, Address Guesser, and Natural-language Map; and
- information comprehensibility (language-proofing technology), which includes Multilanguage Helper and Style Enhancer.

Both groups share the same linguistic engine, a natural-language parser supported by a preparser that deals mainly with named entities, addresses, acronyms, and numeric expression handling. Apart from minor modifications such as the named entities mark-up format, all the tools exploit the same linguistic resources. Also in this case, the distinction between different levels of analysis (each contributing to different parts of the parser’s informational structure) lets developers adopt this solution. For instance, although the information-access tools rely on the analyzer’s lists of triples, the information comprehensibility tools use lexical or structural information associated with phrases (for example, one phrase with several levels of embedding indicates low readability).

We obtained another key reusability result while developing the German grammar. Instead of producing a new grammar from scratch, we based the German version of the tools on general techniques and principles for multilingual linguistic accounts.^{2,3} (John Bateman of the University of Bremen developed the German grammar, and we based the reuse strategy mostly on his research.) We applied such techniques to modify the existing Dutch grammar, thus obtaining the German grammar.

The two languages have a similar word order, and this approach was reliable and effective. We developed the German grammar in less than one person-year, obtaining good precision and recall results.

This experiment's results strongly supported the general validity of employing multilingual development strategies for grammar component reuse across languages. Furthermore, EDEN became a major experiment with APSGs and rule-based, full-path parsers.

Lessons learned

EDEN's NLP tools all use a linguistic-analysis engine constituted by a rule-based parser. Rule-based parsers are generally seen as less reliable and less efficient for real-world applications and less reusable in different application domains than systems based on heuristics and mark-up devices. Although previous research has demonstrated migrating grammars within natural-language families,^{2,4} only a few studies are available for APSPG, and they don't discuss the specific grammar family (that is, APSPG specification) that the development tools apply. The NLP technology that the EDEN tools apply resulted from rethinking and customizing the outcomes of previous computational linguistics research^{5,6} and experience in developing a commercial solution for NLP-enhanced Web search engines.⁷

We initially used an Italian grammar as the first testbed for the approach (which drove the development of other linguistic analyzers, namely English, Dutch, and German) and for integration solutions. This core grammar covered most basic Italian syntactic structures and had already been experimented with in information-retrieval and search-engine prototypes.

We based our initial work before the EDEN project on collections of syntactic structures, and we partially tested the resulting grammar on real-language corpora. This work included a formal evaluation using a small annotated corpus (hundreds of sentences) that returned an approximate average value of 70 percent for precision and recall. EDEN users' tests returned a significant amount of data on the grammar's performance on real texts in the application domain. These data were the basis for grammar maintenance and dictionary extension.

The EDEN tools' technical approach requires that the grammar cover at least one syntactic phrase-level category (such as a noun phrase). Our tests of English and Italian texts demonstrated that applications can return good

results when the grammar covers basic structures of noun phrases, verb phrases, and prepositional groups; adjectival phrases carry useful information as well. Empirically, we found that complete coverage of noun-phrase structures is the critical threshold for obtaining good results. Grammar doesn't need to cover sentence structure; however, according to full-path parsing, sentence-structure recognition can significantly improve the parser's accuracy, cutting out some ambiguity. (Our initial tests with Italian and English texts use grammars that cover nuclear sentence structure—that is, sentences comprising a main verb and its internal arguments, such as subject, direct object, and indirect object.)

We chose a simplified version of depen-

This experiment's results supported the general validity of employing multilingual development strategies for grammar component reuse across languages.

ency-oriented parsers' typical format (see <http://ufal.mff.cuni.cz/dg/dgmain.html>) for the output's format. We wanted to make the grammar compliant to initiatives such as the EAGLES project and, later on, to Semantic Web Organization and W3C RDF guidelines. Although this output format lacked information about cross-references among phrasal constituents, it was extremely flexible and powerful enough for different applications, especially information-retrieval systems. The format didn't undergo significant changes during the EDEN NLP tool development.

Starting with the assumption that the grammar's main purpose is information extraction, we developed the grammatical rules set to focus on recognizing basic constituents rather than covering complex structures. From a linguistic viewpoint, we devoted major effort to covering basic structures for nominal phrases that we generally considered to be most relevant for information retrieval and to covering main verb-noun or verb-adjective relationships. Our final goal was to achieve at least the

same precision and recall as traditional information-retrieval systems, which typically focus on nominal parts, while also taking a further step—considering noun-verb, noun-adjective, and verb-adjective pairs. Incidentally, this approach should guarantee easy replication in different languages because the required syntactic rules set, at least for Western languages, isn't huge. Also, the dictionary structure isn't complex because this approach doesn't require large amounts of lexical information. As a result, we obtained a trade-off between two theoretical approaches generally considered to be opposites: the linguistic analyzer is a rule-based approach, but it doesn't need a complete analysis of input sentences.

One especially relevant result was that the localization of the toolset to different languages didn't require significant changes to the initial design. The EDEN project has thus been a relevant experience in the reuse of linguistic resources. The experiment's particular aspects were threefold:

- The project's four linguistic analyzers (Dutch, English, German, and Italian) were deployed using the same development tool (YAP4NL, or Yet Another Parser for Natural Language; www.tarasi.it).
- Consequently, the analyzers all shared the same linguistic-analysis approach (rule-based, full-path parsing with a post-parsing procedure, simulating a shallow parser).
- Finally, from a software design perspective, no major change or significant integration was necessary for localizing the toolset.

The pilot phase

In preparation for the pilot phase, the partners considered a range of criteria and data-gathering methods to evaluate the desired communication improvements. We don't have the space to describe them here in full, but we can outline the evaluation questions, criteria, and evidence-gathering methods. The evaluation was more limited than we intended, delayed by our efforts to ensure that each pilot site accepted the software tools.

The evaluation

The evaluation sought answers to the following questions:

1. Do the NLP-based tools provide relevant answers to citizens' questions?
2. Do the pilots demonstrate anticipated improvements to online access, navigation, and comprehension?

3. To what extent do the citizen and PA users accept the tools, and why?
4. Do EDEN tools better enable citizens to contribute views on their neighborhoods?

Criteria and methods. Most of our effort went to addressing the first question, which mainly concerned the Answer Tree and Address Guesser tools. We can consider both to be question-answering systems and can assess them as such using the standard measures of precision (that is, the number of relevant answers in the results set divided by the total number of answers returned) and recall (the number of relevant answers in the results set divided by the number of relevant answers present in the system). We'll return to question 1 later.

We had intended the evaluation to focus on questions 2 through 4. Acceptable information-retrieval performance was a necessary but insufficient condition for EDEN's aims. Regardless of whether an information-retrieval system achieved high precision and recall in matching a query's text, end users cared whether the results met their overall information-seeking needs. Furthermore, the users cared whether that led to more-informed public participation in decisions (for example, on local planning proposals).

The EDEN tools aimed to meet the end users' and users' needs by making it easier to access, navigate, and comprehend online information. They also aimed to offer an acceptable online route for citizens to comment on local planning proposals and for policy makers to act on such comments. To operationalize these criteria, technical partners drafted more exact indicators and working definitions and discussed them with the PA partners and "user panels" of citizens.

We expressed the indicators and targets as statements related to each tool and to several methods or sources of data. These were primarily qualitative and included

- interviews and group discussions with officers and citizens invited to user panels of approximately six people;
- online surveys for tool users, presented to PA Web site visitors, which sought satisfaction ratings for accessibility, navigation, comprehension, and acceptability on Likert-type scales (the user partners used a similar approach to assess the quality of discussion forum contributions); and
- log files—the tools logged their usage in terms of queries or comments made and any

responses that the system provided (such as FAQs and office addresses). Web server log files also provided details of page requests and visits to indicate navigation routes.

Returning to information retrieval effectiveness, we noted earlier that this evaluated the recall and precision of searches made using the Answer Tree and Address Guesser tools. The specific measures were *mean recall*; *mean average precision*, which combines elements of recall and precision in a single score; and *mean reciprocal rank*, which gives weight to a relevant answer based on its position in the results set. So, we assessed the reciprocal rank score for each of 100 queries, then calculated the mean. For example, if an

The end users who tested the tools were on average satisfied with them; most said that they normally made inquiries by telephone or in person rather than existing online channels.

Answer Tree query gives a set of three question-answer pairs and only the second and third are relevant, the score is 0 for the first, 0.5 for the second, and 0.33 for the third, giving a reciprocal rank score of 0.83.

We conducted these tests first as an internal validation test to ensure that we had properly set up the data (FAQs for Answer Tree, office email samples for Address Guesser) to get the best results from the NLP software resources. Because our rationale for using NLP was to help ordinary citizens without technical skills search effectively, our validation targets were high. For example, for Answer Tree, the target for mean reciprocal rank was 0.87 (that is, 75 percent of results for test queries should show a relevant FAQ first, and the rest should show one in second place).

To perform the validation tests, each pilot site

- wrote 100 queries to cover their Answer Tree FAQs and the offices involved in testing Address Guesser;

- identified each query's expected results (FAQs or offices), listing them in a spreadsheet; and
- fed each test query into their software for each test run.

Then, Napier University retrieved the results, matched them against the expected results in a spreadsheet, and calculated the results for each query and the test run average.

For the evaluation, we repeated these tests using a sample of queries that end users had entered online during the pilots together with the responses, which we extracted from the log files.

It's important to note that such tests are normally conducted by information-retrieval specialists in laboratory settings (using test-query sets that the specialists compile to test the system's capabilities) rather than by PA users (albeit with the remote guidance of technical partners). The test queries' composition was a critical issue because it highlighted differences in expectation between the testers (user partners) and the technical partners. These differences came to the fore when we compared the results from prewritten queries with those from the end users' real queries. The targets for evaluation with real users' queries were lower, even though the internal validation tests gave positive results.

Results. The online questionnaires, interviews, and comments from users and end users indicated success or partial success on four of the six indicators for Answer Tree and Address Guesser (further details are available at http://itc.napier.ac.uk/ITC/Documents/EDEN_D62_Summary.pdf). The end users who tested the tools were on average satisfied with them; most said that they normally made inquiries by telephone or in person rather than existing online channels. One important finding was that most end users didn't participate in city-planning consultations by traditional means such as public meetings. Overall, sizeable minorities of the end users agreed that the tools better prepared them to contribute their views online. However, the PA that piloted the Style Enhancer found that it offered insufficient added value to the editorial team (the intended end users).

We can see how acceptability for e-democracy purposes depends on information-retrieval effectiveness by considering just one tool function. Answer Tree gives end users an email route to a general inquiry-handling office, the Administrator. A key design

assumption is that citizens would use this email facility only when they can't find a relevant answer in the FAQ tree—that is, for unusual questions that may be useful raw material for new FAQs—thus alleviating the inquiry-handling officers of the more routine ones. However, if the search function has low performance, there is a risk of the Administrator being inundated with emailed queries—representing a risk of failure on the acceptability criterion.

The mean average precision and reciprocal rank tests were therefore crucial to establishing that the Answer Tree and Address Guesser tools worked before launching the pilots (validation tests) and that they worked well enough with the end users' queries during the pilots (evaluation tests). These tests came to two conclusions.

Answer Tree was better at retrieving relevant FAQs in response to natural-language queries than the widely used SWISH-E (Simple Web Indexing System for Humans—Enhanced) indexing and retrieval algorithm. In fact, we designed Answer Tree to run alongside SWISH-E, and we found the difference in performance by repeating tests with the same queries. The evaluation target was quite modest. For citizen queries that contained grammatical errors, we didn't expect the NLP parser to perform well. So, the target was that a matching FAQ should almost always be one of the first two results and at least half the matching FAQs should be shown (a mean reciprocal rank of 0.5 for 90 percent of queries and 50 percent mean recall or mean average precision). The normal operating mode (with the NLP parser) exceeded the first target (0.65 mean reciprocal rank) and almost reached the second (43 percent mean recall). Moreover, these were 24 percent and 2 percent more than we achieved when using only SWISH-E.

Address Guesser results weren't accurate enough for users to be confident that the guessed addresses were correct. Like Answer Tree, the targets were that a correctly guessed office should almost always be one of the first two results and at least half of the correct addresses should be shown. Again, this corresponds to a mean reciprocal rank of 0.5 for 90 percent of queries and 50 percent mean recall or mean average precision. The actual results were 0.35 and 37 percent. However, we thought that refining the training samples and the interface design would likely improve performance (we discuss how we achieved this in the "Consequences for testing and deployment" subsection).

Exploiting NLP in a specific domain

Perhaps the most ambitious task was exploiting advanced NLP technologies (normally used for experimental purposes with highly structured domains) in a real-life context and a loosely structured domain, such as urban planning. A detailed description of that domain is outside this article's scope. However, there's a statutory need for PAs to consult the public on urban-planning matters, and standard operating procedures exist for doing so, making the domain suitable for e-democracy experiments. Also, online communications between citizens and the PA on planning matters were in their infancy, so we lacked a well-established corpora of email communi-

Given the project's collaborative nature, technical risks, and pressure for quick deployment, the project partners knew that they required a shared understanding of each other's needs.

cations, relevant online publications, and comparable search and retrieval facilities from which to extract previous queries. The existing prior examples were highly variable in terms of text length and syntax. Finally, urban planning has a wide lexicon of technical phrases, including polysemous terms (with several meanings).

As we've already mentioned, the approach was risky. Alternative approaches to facilitating information retrieval were possible, and providing improved information wouldn't necessarily result in more (or more informed) participation in online dialogue about planning proposals. Much depended therefore on the user partners' ability to quickly demonstrate improvements in the online information-retrieval capabilities for nonspecialists.

Given the project's collaborative nature, technical risks, and pressure for quick deployment, the project partners knew that they required a shared understanding of each other's needs. On the one hand, the user partners knew that the NLP technology fell short of natural-

language understanding, limited as it was to syntax analysis. They also recognized the necessity of training and fine-tuning the software. However, their focus was largely on agreeing on criteria to evaluate the tools' effectiveness and building internal support for the project's pilot phase—a necessary effort because the capabilities and commitment to deploy the tools spanned a range of PA departments. Similarly, the technical partners focused on continuing to develop the tools in the face of continued changes in the PA business environment. Despite, or perhaps because of, the shared focus on meeting changes in the functional requirements and the project's political environment, the partners considered testing the software's efficacy with a suitable test set of data to be vital but relatively straightforward.

Setting up and testing the tools involved considerable effort and some conflict over the definition of *natural language*. The project partners widely understood this to mean the set of all languages humans use to communicate, as opposed to formal languages such as logics or programming languages. For the technical partners, however, *natural language* had specific connotations beyond that definition.

In the syntactic-processing approach, an NL parser aims to determine whether a phrase or sentence belongs to the language at issue and, if it does, to assign a syntactic structure to that sentence. So, the parser applies the rules within the grammar: if the analysis of a phrase or sentence lets the parser build a higher node, that phrase or sentence belongs to the language. This identifies the phrase or sentence as well-formed. So, the technical partners understood natural language to comprise the theoretical (infinite) set of all well-formed phrases and sentences of a language humans use to communicate. The user partners understood natural language in the more general sense: the language used by end users, regardless of well-formedness.

These differences were reasonable given the partners' different professional backgrounds, but they had huge consequences for deploying and testing the NLP tools. The technical partners thought it was obvious that a tool's effectiveness and performance are highly dubious if it must consider all possible syntactically ill-formed phrases and sentences. The user partners thought it was equally obvious that email communication often contains grammatical mistakes and spelling errors; indeed, this was so obvious that they didn't ask for a spell-checking preprocessing func-

tion until late in the project, and they had assumed that the NLP approach could cope with such lexical and syntactic errors. The technical partners, on the other hand, assumed that the user partners would recognize that to handle such errors as effortlessly as humans do, the NLP technology would need semantic and pragmatic capabilities that the project wasn't meant to address. This understanding gap highlights a technical challenge when using NLP in free domains and control-free environments (such as EDEN): balancing the necessity for the software to both enforce syntactic rules to process well-formed texts and also allow for users who might write texts that aren't grammatically well-formed.

Consequences for testing and deployment.

As we mentioned earlier, the Answer Tree evaluation tests exceeded our targets. These tests had real queries, including those that weren't grammatically well-formed. We excluded some queries because they didn't link conceptually to the urban planning domain (such as the question "Where can I register my dog?"), while others were too specific to fit an FAQ page (such as "Where may I park on Main St. when coming from my house?").

Because Address Guesser fell short of its targets, we needed to refine the data used to train this tool. Testing showed that the NLP technology was effective when the system was completely set up, but reaching that level can be resource intensive.

The Address Guesser training phase involved the following: each office that the tool must match to an end user's query (entered as a Web form-based email text) is defined on the basis of a certain number of manually assigned messages that characterize that office's activities. The messages had to be handled by domain experts (that is, PA personnel familiar with the task of routing inquiries to the appropriate office).

The first experimentation gave very poor results. The understanding gap meant that the personnel conducting the training didn't check the real messages they collected apart from ensuring that they were made anonymous for privacy reasons. The test messages were assigned to the various offices without verifying their linguistic consistency in terms of syntactic (and spelling) well-formedness, without deleting possible noise factors (such as introductions and greetings) or proportionately distributing the messages across offices.

We subsequently addressed these issues

through improved human training materials but couldn't fully address the consequences for the software—namely poor general performance and a performance bias toward some offices due to the disproportionate assignment of messages.

Answer Tree wasn't immune to such problems despite its better performance. Performance increased dramatically after we released a manual for FAQ list editing and introduced a stop-words list (that is, a list of words for the system to ignore) appropriately compiled for the application.

Grammar and style-checking with rule-based systems. Certainly the tool that showed the greatest technological limitations was the Style Enhancer. Its analysis and suggestions concerned both *lexical complexity* (the use of technical and jargon terms) and *syntactic complexity* (structures traditionally considered too complex for easy reading).

Style-checking derives from so-called controlled languages. In rather small, well-structured domains (such as the development of software functional specifications expressed in natural language), it's easy to define rules to identify mistakes and errors that must not occur in the document. (One classical example is that system performance specifications must never use the word *should*; they should instead use less vague expressions). This lets you eliminate ambiguity and vagueness as much as possible so that any programmer can read the final document and unequivocally interpret it. It's clear that having this kind of control on functional specifications documents is necessary, especially in large environments, where programmers often can't physically interact with the specifications author. On the other side, this necessity has become less urgent thanks to the introduction of representation languages for functional specifications that are both simple and unambiguous, such as UML.

Rules that can be used for controlled languages don't completely fit with a less-structured domain such as urban planning, where language is both free and complex. EDEN adopted rules that were mainly based on linguistic common sense (for example, two embedded clauses in a single sentence are more complex than two simple sentences). But although quantitative linguistics has produced good systems over the past years, EDEN didn't produce enough urban planning corpora from which we could extract a significant number of rules.

In the case of the Style Enhancer, PA users had many misunderstandings about the tool's nature and usefulness, often taking it for a grammar checker. A grammar checker doesn't check a sentence or document's readability but verifies its grammaticality. Commercial products usually have reduced functionality and are limited to checking the morpho-syntactic agreement among the sentence's elements. To signal (often meaninglessly) the presence of certain structures, commercial products suggest, for instance, not to use the passive form (as in some versions of MS Word grammar checker).

We can summarize EDEN's aim with the motto "bringing technology to the people." Nonetheless, it's difficult to properly complete such a task when those people care more about results than how the technology works. Most just didn't care that a given system was based on bottom-up parsing techniques integrated with a simulation of a top-down filter, or that it includes an optimization algorithm that packages the analyzer-generated parse forest, greatly improving the analysis. Most likely, the end user didn't care to know triumphant data about precision, recall, or mean reciprocal rank. The end user simply wanted a friendly, reliable system that saves time and energy.

Private companies' marketing personnel will solve this problem without frustrating technicians too much. It's worth stressing, however, the consequences for our *management* of technologies when deployed in e-democracy.

A major problem we encountered concerned how well-formed the users' input was. Our NLP's theoretical backbone was the assumption that the language fragment we investigated shows some (minimum) criteria of grammaticality. On the other hand, many citizens don't edit syntactic mistakes from their email. Does this mean that NLP, by definition, can't manage the language citizens use in their email? No. It means, instead, that NLP technical people must thoroughly analyze a significantly large corpus of the language and calibrate their instruments until these can handle linguistic objects that fall outside their field of interest.

Clearly, this doesn't mean that we should start thinking about a system designed to parse anything, but we should start investigating and using instruments that let us analyze poorly formed sentences (as EDEN did for some sen-

tence-level mistakes). For instance, adopting shallow parsing strategies to analyze meaningful fragments in ungrammatical sentences, or loosening certain rules contained in the grammar, would improve performance for a system dealing with email language. This means moving toward a sort of DNLP (Dirty Natural Language Processing), which allows us to exploit all of NLP's theoretical advantages and at the same time successfully manage an imperfect language. This was one of EDEN's greatest challenges, and it will arise whenever we use NLP in e-democracy. The new motto, then, could be "bringing technology to the people without letting them know."

EDEN's structure has from the beginning highlighted one weakness of the adopted NLP-based solution: customization, set-up, and maintenance work require skilled people with specific profiles that aren't common in the IT world.⁸ According to the project plan, some PAs were in charge of developing language-specific resources (such as grammars, lexicons, and support tools or databases) and setting up localized versions of the toolset. These cities encountered major difficulties both in finding people of the requested profile who could participate in the development and also in setting up the needed infrastructure (work groups, databases, and servers).

At the same time, NLP technology hasn't yet reached the same stability or maturation as other information-processing techniques. This could cause difficulties in communicating ideas and cooperation, and it could keep linguists engaged in software development from quickly agreeing on the system architecture and, later, from easily identifying weak points and solutions.

A major positive side effect of EDEN for the developers was an appreciation of the perspectives and possibilities that open source software policies offer. These appeared to the technical partners to fit linguistic engineering solutions' development environment. They could provide a constitution to support the community of technicians and users whose cooperation is necessary for linguistic engineering technologies to grow and become more stable. This support becomes even more critical when facing the broad needs that e-democracy represents, and it seems reasonable to approach e-democracy by seeking a democratic approach to software solutions. ■

Acknowledgments

EDEN was EU Commission cofunded as project IST-1999-20230.

References

1. J. Bateman, *The Re-Use of a Dutch Augmented Phrase Structure Grammar for German: Status Report and Prospects*, EDEN tech. report, 2002.
2. J. Bateman, "Enabling Technology for Multilingual Natural Language Generation: The KPML Development Environment," *J. Natural Language Eng.*, vol. 3, 1997, pp. 15–55.
3. M. Rayner, D. Carter, and P. Bouillon, "Adapting the Core Language Engine to French and Spanish," *Proc. 1st Int'l Conf. Industrial Applications of Natural Language Processing (IANLP)*, 1996.
4. J. Pinkham, "Grammar Sharing in French and English," *Proc. 1st Int'l Conf. Industrial Applications of Natural Language Processing*, 1996.
5. I. Prodanof et al., "A Grammar Development Environment for Reusable and Easily Customizable NL Applications," *Proc. 1st Int'l Conf. Language Resources and Evaluation, ELRA*, 1998, pp. 611–617.
6. I. Prodanof, "Resources, Tools, Reusability," *Proc. VEXTAL*, 1999, pp. 9–15.
7. M. Carenini and M. Vanocchi, "Managing Documents through the BRAIN Technology," *Proc. 7th Bar-Ilan Int'l Symp. Foundations of Artificial Intelligence (BISFAD)*, 2001.
8. M. Carenini, "NLP between Research and Production: From BRAIN to MIND," *Linguistics and the New Professions*, A. Giacalone and E. Rigotti, eds., Franco Angeli, 2003, pp. 127–139.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

The Authors



Michele Carenini is the international business development manager at NoemaLife. He was also the EDEN technical coordinator. His research interests include feasibility of real-world AI systems, knowledge management, and natural language semantics. He received his MS in computational linguistics from the University of Pisa. He's a member of the Society for Machines and Mentality. Contact him at NoemaLife SpA, v. de' Carracci 93, 40131 Bologna, Italy; mcarenini@noemalife.com.



Angus Whyte is a senior research fellow at Napier University's International Teledemocracy Centre, where he evaluates e-democracy systems. His research interests focus on combining qualitative methods including ethnography and action research to explore innovation in governance and its implications for public participation. He received his PhD in information science from Strathclyde University. Contact him at the International Teledemocracy Centre, Napier Univ., 10 Colinton Rd., Edinburgh EH10 5DT, UK; a.whyte@napier.ac.uk.



Lorenzo Bertorello works at the EU Affairs Department of Liguria Regional Government. He was also the EDEN project manager. His research interests include EU scientific policies and programs, e-democracy and e-government services, geographical information systems and applications, and public administrations' innovation through new technologies. He received his law degree from the University of Genoa. Contact him at EU Affairs and International Relations, Liguria Region, Piazza De Ferrari 1, 16121, Genova, Italy; lorenzo.bertorello@regione.liguria.it.



Massimo Vanocchi is a consultant for the Lexical Resources and Tools for Italian project, the Institute for Computational Linguistics of the Italian National Research Council, and Regulus SpA's e-Ten HANDS project. He received his MA in applied linguistics from the University of Pisa. His research interests include R&D, computational linguistics, NLP, syntactic parsing in different application domains, theoretical linguistics, text quality evaluation, and indexing technologies for search engines and document management systems. Contact him at via Bengasi 67, 57012 Castiglioncello (LI), Italy; m.vanocchi@sytwo.com.