

# Technology Transfer: Some Observations from the TIPSTER Text Program

*Dr. Sarah M. Taylor*  
Office of Research and Development  
Washington, DC 20505  
E-mail: sarah@ucia.gov  
Telephone: 703-351-2565

## 1. Introduction

Technology Transfer has been an important part of the TIPSTER Text Program from the beginning. Research alone was insufficient as a motivation for the program. What was discovered in the laboratory had also to be transferred as quickly as possible into the workplace. The central role of technology transfer in the initial formation of TIPSTER had several causes. Government sponsors and initiators of the program, in 1990, could see clearly the inadequacies of the tools that analysts were working with, at the time, and could also see that the overload of text which analysts dealt with was only going to get worse, given the proliferation of information sources and the increasing push in the Intelligence Community to tap those sources. Knowing the normally long time it takes research advances to make their way into standard technology, the founders of the program rightly believed a concerted effort to place good technology in the hands of users would be necessary to insure that they received the benefits of the program as soon as possible. There were other pressures: continuing reductions in the numbers of analysts available to do analysis, reductions in budgets, and constant pressures to justify Government sponsored research in terms of demonstrable and practical benefits to the Government.

The research fields sponsored by TIPSTER are application oriented in any case. Both Document Detection and Information Extraction imply an eventual user - someone who needs documents or someone who needs particular information about particular kinds of entities or events. Successful research does thus have direct implications for operational applications. But more broadly, applications can be important to much research that deals with human language. Language exists for the communication of meaning. Any automated processing of language cannot be really evaluated outside the context of the actions of expressing and understanding, both actions which are highly situation, or task, dependent. In addition, language is a human construct and, whatever its imperfections, a person is

the only real authority we have on language, the only measure of whether or not meaning has been conveyed. The application of automated language processing to concrete human tasks is itself an important research method in this field. Although the goal of strengthening the science through the application of the technology was not explicitly stated as a reason for the emphasis on technology transfer in the TIPSTER Program, nonetheless I think the emphasis on tasks and on the usefulness of the technology is benefiting the underlying science of computational linguistics.

Central to the initial TIPSTER planning, then, was the goal of technology transfer. The magnitude of the difficulty of achieving that goal was perhaps not well understood in the early days of the program; but since then, we have all learned that this transfer does not happen automatically, but requires even more effort, planning, and creativity than the research. The goal of technology transfer is subscribed to by both Government and contractor participants. There are considerable potential rewards which serve as important motivations to people involved in the process - material incentives, but also the satisfaction of building something that works and does useful tasks for people who need it. These rewards are important, because the process of technology transfer is difficult and messy. It requires the persistence and flexibility to tackle many obstacles. It is as much a matter of business and psychology, as it is of technology and engineering.

TIPSTER Phase II has made a number of strides forward in transferring the research advances of Phase I into operational use. Collectively the participants in the program have learned a great deal about what works and does not work in transferring technology. The program remains committed to continuing an aggressive technology transfer effort. This paper summarizes what I have learned from the Phase II effort. It can perhaps serve as a basis for others to reflect on the same issues. A record of what was learned in Phase II efforts will be valuable for continued tech transfer in the future.

*This material has been reviewed by the CIA. That review neither constitutes CIA authentication of information nor implies CIA endorsement of the author's views.*

## 2. What is meant by Technology Transfer?

In the TIPSTER Program the phrase “technology transfer” means taking algorithms or methods which have been developed in a laboratory setting and proven themselves potentially valuable, through the evaluation process, and making these algorithms available for use by analysts in their own working environment to do their job, on a daily basis. This implies not only technology which can work fast enough, without breaking, to meet operational needs, but also technology which in most cases is reasonably well integrated into the analyst’s existing work environment, which has a decent human interface, which provides value added to the analyst, and which can be operated and maintained by the user’s organization.

There are other possible meanings for this phrase. It is used, in some contexts, to refer to the process of transferring the responsibility for development of a technology from one group of people to another, particularly as the development changes stages; for example, the transfer of technology developed under Government sponsorship, in a research lab, to private sector sponsorship, for commercialization. The phrase is used in other contexts to mean the transfer of the technology *and its mode of use* from one set of users to another; for example, the sale of advanced weapons systems from the United States, or any other possessor of such systems, to a country which does not yet use such weaponry, is a technology transfer operation.

The use of the phrase in the TIPSTER Program, but not unique to it, broadens and combines these two meanings. In TIPSTER, the goal has been not only to get the newer technology into the hands of specific users who need it, but also to make the technology widely available within Government. The technologies researched in TIPSTER have application to text handling tasks throughout the Government and applying it successfully in one or two well defined places would not be sufficient to meet the needs which the program set out to address. Immediate transfer to users, with an articulated mode of use, addresses some of these needs. However, it is also necessary to include transfer of the production of the applications of the technology to a sufficiently broad spectrum of vendors so that the technology can be made available to many Government customers. In the TIPSTER case, technology transfer has meant the commercialization or productization of at least some of the research advances, and the spread of a reasonable number of the better ideas from the Government sponsored research into a broad range of products, as well as back into new research projects.

A close look at the various uses of the phrase “technology transfer” makes clear that the transfer process is not one of simply handing a *thing* from one group of people to another. *Things* are indeed transferred, but knowledge must go with the *thing*. This transfer of knowledge is one of the major challenges in the process - knowledge of how something works, of how to build it, integrate it, maintain it, improve it and most of all, how to use it. The knowledge of how something works and how to build it is presumably a natural by-product of developing a technology in the first place. The knowledge of how to integrate, maintain and improve it, is more difficult to come by, but is still within the standard domain of applications developers. However, knowledge of how to use these technologies is much more difficult to develop and is one of the major stumbling blocks to transferring the technology to more general use.

## 3. Some obstacles to easy transfer of TIPSTER technologies

Many of us were probably taught the adage, at our mother’s knee, “build a better mousetrap and the world will beat a pathway to your door.” Implication: an entrepreneur need only get a good idea, for something that people need, and build it well. The rest will take care of itself. Excellence will be recognized and rewarded by the forces of the market place. Turns out “it ain’t necessarily so,” depending, perhaps, on how one defines “better mousetrap.” This section describes some of the problems that we have encountered in transferring technology into useful applications. Some of the solutions we have employed to these difficulties are described later in the paper.

### Getting technology to specific users

Finding people to try out a technology can be surprisingly difficult. A number of pressures can deter potential users from trying a new technology. Helping with a development and learning a new set of procedures takes time and energy. In the highly stressed environment of many Government users, who are overworked, working in barely comfortable environments, and may be using a plethora of badly integrated or unintegrated computer systems, this added demand may seem impossible to meet. It is very difficult for users to judge ahead of time whether and how much the technology will actually benefit them. Most do not understand how the technology works; they must take it essentially on faith that the technology will do what it is supposed to do, and that what it is supposed to do will actually help them. This is particularly true for the TIPSTER technologies of Document Detection and Information Extraction, in

which processing is almost entirely hidden from the user. These two capabilities often sit inside other more visible processes, so it is not always clear to the user what added benefit they bring.

Understanding the user's tasks and developing a good technology match for them is the first necessity for an application. More than superficial understanding of the user's job is required, even to simply explore the possibility of an application. The developer should understand the "guzintas" and the "guzoutas" in some reasonable detail, even to make a proposal to a user as to how the technology might be applied. But how is the developer to get that knowledge? The user has no time to discuss his problems in detail with anyone. The user may not be able to describe his work flow with sufficient abstraction and accuracy so that a developer will be able to understand what is required. Developer and user come from two different worlds; in most cases, use vocabulary that is misunderstood by each other; approach technology differently; and, approach analytical tasks differently. Communication between them can be poor and extrapolation by the developer to fill in what is not told can be inaccurate. In addition, there may be security restrictions which further complicate the process of understanding the user's data and tasks.

TIPSTER technologies, by their very nature, will be integrated into a larger system. Much of this system consists of further hardware and software, sometimes highly outdated. But the system *always* includes the user and the user's work flow. Part of understanding the user's task is understanding not just the specific Detection or Extraction events that may have to take place, but understanding how these fit into the entire process the user goes through to get his job done. Information about the components with which the TIPSTER technology must interact can be difficult to obtain. The technical solutions to the integration into legacy systems may be awkward, at best. Existing systems can be poorly documented and consist entirely of proprietary code. They may be old enough that modularity and open architecture were not common practices when they were developed. There can be security restrictions on information about how they work. The environment itself can be changing, with constant upgrades to operating systems and basic work packages; if the hardware/software environment is not being well managed, integrating into it can be a nightmare.

Besides the end user - that is the user that will process text information with the application - there are others who use an application and who must be accommodated in the development of an application: the people who will support and maintain the system.

These include system administrators and system maintainers. While these people are closer to the developers in their experience of technology and thus may communicate reasonably easily with developers, they may still resist the trial of new technology for other reasons. The new technology, unless a supported COTS product, may seem too risky in terms of both integration and maintenance. It may require hardware and software packages with which the support personnel are not familiar and for which the larger organization does not provide support. They may be afraid that the new system might completely overload existing network or indexing capacity, which problem could not be solved without affecting many more work groups and applications than the targeted group. The support staffs themselves may not have the dollars or personnel hours to provide for any systems other than the ones they already maintain. It is difficult to project the costs of maintenance of a yet unbuilt and untested system, and support staffs are justified in remembering their worst experiences. They rightly do not want to get involved in costly maintenance contracts and can be wary of anything which smacks of a proprietary system, requiring maintenance agreements with a single vendor for years to come. Additionally, the support staff may be largely seeing the *cost side* of the new technology, without receiving much from the *benefit side* of the greater functionality.

Of course, there are also the issues none of us like to talk about publicly, like bureaucratic politics and turf battles. New technology can pose a challenge to existing equities. Users may not want to share with others the problems with their current work methods, as these might suggest there are inadequacies in their product. Some users may be afraid that the new technology will put them out of work. Some may be afraid that the technology will only bring them more work. One set of users, offered new Detection technology, with the promise that it would provide them with more relevant documents, stopped listening at that point and announced they did not want to get more documents; they already had as much as they could read. End of discussion. There was no chance to explain that the technology could help them get to the best documents more quickly and thus, in the end, save them effort and allow them to do better work. Existing technology may be defended by managers who made the decision to buy and use it; it may be defended by the contractors who installed and maintain it. New technology, provided by non-users, can be seen as a threat to the user's autonomy and authority over his task: "no-one can tell me, or even help me, do my job, except maybe someone who has done, under the same stresses as myself, the same kind of job."

It can be difficult for the person responsible for transferring technology to come to agreement with the

user group on the appropriate scope and goals for a project. Users who are eager to try new technology may be unsympathetic with some of the baby steps necessary to make technology insertion go smoothly. Many of us are familiar with the user who sees a prototype, likes it, insists on keeping it, and then is disillusioned when the software proves not to be robust or the throughput is insufficient. Most projects seek to make an improvement to an existing process, but that can be difficult to determine since old methods of doing the work have not been baselined. While processing speeds and throughput of established systems are known, quality of output, accuracy of the automated processing steps in terms of Recall, and some measure of the ease of use of established systems are not. Generally, the collective will to baseline the quality of older processes does not seem to exist. Perhaps people feel that the systems must be upgraded in any case to newer hardware and software suites - they will do the best they can to get the best new system and not worry too much about how good or poor the old system was. This means that any improvements enabled by the new technology cannot be established objectively. Improvement will be entirely in the perception of the various users of a system and anyone who uses the results of their work. A new system could provide many benefits, but if for some reason the user does not like it, the benefits may not be perceived or may not be sufficient to outweigh the negative aspects of the new system. This puts a heavy burden on the purveyor of new technology, to manage its insertion in such a way that it *both* pleases the users *and* does them good, not ill.

### **Making technology broadly available**

Meeting the second part of the TIPSTER technology transfer goal, that is making the technology widely available, at least to the Government, at reasonable costs, poses further problems. Commercialization of TIPSTER research may provide some answers. However, the technology can be costly, even if commercialized, in terms of licenses, integration costs, and maintenance costs. Whether the applications are handled by a commercial firm or not, the market for advanced technology in this area may simply not be large enough to support lower costs.

The goal of broad availability of the technology at reasonable cost provides one potential area of conflict between the Government sponsors of the program and developers who work with us. Some developers may be interested in commercializing products and others may be interested in building custom applications for specific customers. Both groups can have an interest in keeping their technology advances secret in order to improve their competitive position, thus taking a

stance in conflict with the Government's push to make advances widely available at a reasonable cost. In fact, it is highly probable that some competent researchers and developers have not participated in the TIPSTER Program because of a wish to retain complete control over their research advances.

### **Some solutions**

Many of these problems are not unique to TIPSTER and potential solutions to these problems can come from a wide variety of sources. TIPSTER applications can by no means claim a perfect record in meeting all of these issues and resolving them. However, in the Program and elsewhere we have made a lot of progress in learning how to deal with them.

Many of these problems stem from the fact that TIPSTER is working in a relatively new technology area. Automated information systems are, in the history of technology applications, a comparatively new phenomenon. Many of the issues that are unclear or rapidly changing - such as the way in which the technology will be used, or the hardware/software environment - are so simply for this reason. Such problems cannot be made to disappear; they must be, instead, accepted and dealt with as best as possible. Other issues - such as the stressed work environment of our users - also are not under our control. However, there are steps we can take to ameliorate the effects of these issues. And there are some problems for which we can offer more complete solutions, because they are more completely within the purview of the TIPSTER applications.

## **4. Understanding the steps in technology transfer**

We know by now that taking technology from the laboratory and getting it to the point where people are using it regularly is not simple. It can help to understand the process to look at it as a series of smaller steps. No one of these steps can be skipped or ignored, although some can be compressed and the entire process can be speeded by the application of more effort at various points and by careful planning.

*The process begins with the research advance and the idea that this particular advance could be helpful to solving a particular user problem.* In the case of TIPSTER technologies, of course, the earliest realization of the utility of Document Detection and Information Extraction to the Intelligence analyst obviously predates the program. The important contribution of the TIPSTER Program was the realization, in 1990, that the two technologies required substantial improvement to be of better use to the

analyst, and instituting a method for effecting that improvement. MUC [1] and TREC [2], the evaluation conferences for the two technologies, became important in helping to determine whether advances were sufficient to take the next steps toward moving the technology into the work place. Given the nature of these evaluations we have not had absolute measurements of the degree of improvement of the TIPSTER technical solutions over the technical solutions that exist in the workplace, but we can show with certainty that individual systems are improving rapidly, and we have strong evidence that they offer better accuracy than most in-place systems.

Another aspect of the early stage of the transfer process, for TIPSTER, was determination of a fairly broad set of new functions on which to work. We came to recognize, as we discussed user needs, that "one size does not fit all" and that part of the problem of existing text handling systems was inflexibility and single focus. Therefore, a program - which had the potential to become a competitive search for single best solutions in both Document Detection and Information Extraction - became instead a search for a variety of solutions, none of which is perfect, but all of which have different useful features to offer. The TIPSTER Architecture was developed partly to give user organizations as much flexibility as possible in implementing solutions. Providing for this flexibility at the early stages of a technology improvement program is necessary, since user requirements cannot and should not be determined in detail at the early stages. A research effort aimed at a very specific set of user requirements would have failed, because the requirements would have changed by the time the research was ready for transfer. In addition, it would have produced solutions which were as single threaded as existing ones. Better technology is itself going to change requirements and should be developed from the beginning with as broad a base of functionality as possible, in order to meet the new requirements that it will generate.

*During the second step in the technology transfer process the technology is made integratable.* When new functionality has been proven to work against either actual analyst data or something extremely close to it and judged to work sufficiently well that it is probably worth the pain and cost of integration, then it is time to begin moving the technology closer to the user's environment. In the TIPSTER Program, this occurred with the inauguration of Phase II. In our case, one important step was to make the technology compatible with a number of possible target software/hardware environments. The greater the number of end environments that can be accommodated the more likely that the technology can be integrated quickly, when a final determination is made to deploy

it. Other aspects of the technology were also addressed, such as the speed of various processes, memory and storage requirements, modularity, interfaces to some standard COTS products, and so forth. For TIPSTER, a major component of the second step was the development of the software architecture, that is the development of accepted definitions for major components, and some modules and interfaces. Independently the demonstration projects also addressed other integration issues, such as the porting of modules from LISP to C and C++, the speeding up of indexing processes, and the integration with standard data base products. In the case of TIPSTER, many of these improvements were paid for by the Government components who needed them. Some were initiated and supported by the individual vendors who saw it to their advantage to provide a more easily integratable product. Outside of the development of the Architecture, only a small amount of integration work was supported directly by the TIPSTER Program Executive Committee.

*The third step is to integrate into an experimental environment so that complete system flows can be demonstrated and important processing threads can be carried out from start to finish.* This step tests the effectiveness of step two and prepares the way for step four. For TIPSTER, demonstration projects undertook this step individually in Phase II. For Phase III, the Architecture and Capabilities Platform will provide an environment to work on these issues. The Free Text Management (FTM) Project [3], for the National Drug Intelligence Center (NDIC), at the Federal Intelligent Document Understanding Laboratory (FIDUL) is an example of a Phase II project devoted entirely to the experimental integration of TIPSTER technologies into a near user environment, for the purpose of engineering and trade-off studies. TIPSTER pilots, such as CANIS [4], to some extent perform the same purpose, although a less broad range of issues can be experimented with under CANIS than in FTM.

*In step four, the deployment in the experimental environment is used to develop a plausible concept of operations in terms of work flow; a plausible user interface is developed to go with it.* CANIS and FTM are both examples of this step. They employ early stage user interfaces for the purpose of investigating the way users would actually like to use the technology. Information from the test and use of these interfaces will be fed into specific requirements for further interface and work flow development if either system is actually going to be moved into full-time operational use. For this reason, at this stage, the interface should be easy to develop, easy to change, and familiar in some aspects to what users know already. A number of demonstration projects use Web browsers for this purpose. In step four, potential users

of the technology are exposed to some of its possible uses, as much as possible seeing their own data flow through the system. Their reactions and suggestions are collected as important sources of ideas about how to actually configure the technology against their task. At this point, it is particularly important for a developer or someone who communicates well with developers to become intimately familiar with the tasks of a number of potential user groups. This person can then experiment in the use of the technology to do these tasks. It is important to have input from users, who may also experiment with the use of the technology if they are inclined to, but it is also vitally important to have someone, familiar with the technology and its potential, look at new ways of tackling the user's issues.

*Step five is to respond to a real application opportunity.* If the preliminary work has been well done (i.e., steps 1-4) the development of an actual application should be relatively rapid, given a reasonably stable environment to integrate into. Rapidity of insertion of the technology at this point is an important objective. Fundamentally, this is because the more quickly a project can successfully be put into place, once users have signed onto it, the more likely it is that the project can avoid a number of serious pitfalls: changed target environment, changed user task, user reorganization, changed user priorities or goals, radical improvements in the base technology which have not been integrated into the design, and perception by the user of excessive amounts of time being required for the project. However, it cannot be stated strongly enough that appropriate preparation (via steps 1-4) must have occurred in order for rapid insertion to take place. Attempts at rapid insertion without the needed preparation leads to many problems, such as applications that are not robust, poorly planned user work flows, poor estimates of integration times, and badly prioritized or developed requirements. These in turn can lead to systems that are not used, to dissatisfied customers, and to heavy reworkings of the system under the guise of O&M (operations and maintenance).

In step five, the use of a standard form of project management cycle becomes appropriate. Requirements for the particular application must be understood, defined, and baselined, accounting for: the user's current work flow and possible changes to it; the target software/hardware environment; the data input, output, and storage formats; maintenance and support; documentation and training; testing and evaluation plans. Rapid prototyping of interfaces, including the GUI, has been successfully used, in ADEPT [5] for example, to help define the requirements and to flush out technical problems. Prior to installation and use, testing and evaluation in an exact replica of the user's

environment is usually necessary. Training for end-users, managers, and support personnel should accompany installation. Further evaluation after installation can determine whether the system has in fact produced the expected improvements.

*Step six is to provide initial system support, rapid responses to problem reports, and as much hands-on user support as possible.* No TIPSTER system with which I am familiar has actually reached this stage as of the writing of this article. However, some general comments are possible from my observations of other systems developments. Customer service, as many commercial organizations are aware, makes the difference between success and failure in many instances. The way in which a system is presented during training and supported will make a big difference in how well it is liked. At the same time, to avoid runaway cost inflation on a project, there must be clear delineation between fixes and changes at this stage. Despite all the careful testing, it is inevitable that there will be some bugs in the software and these must be fixed immediately. However, users can take this stage as an opportunity to change the actual functionality of the software, in effect to change the requirements. While perhaps some of these changes can be responded to, if the budget allows, they are best tracked separately from fixes to the software. A clear understanding between the developer and the user organization concerning how much effort is being put into fixes and how much into actual modifications to respond to a change in the requirements is very helpful. Or there can be an agreed upon period in which changes are made up to a certain level of effort or cost. Finally, within six months to a year from the integration of the technology, the maintenance and support of the system should be transitioned to the user's organization.

## **5. COTS, GOTS and the alternative**

One of the issues the TIPSTER Program has had to struggle with is the form in which its technology can best be provided to the Government user. There is considerable hope among some in Government that many text handling needs can be met with commercial products. It is hoped in this manner to control the cost of using these technologies. If this is to be the case, then TIPSTER technologies would have to become part of standard commercial offerings. The TIPSTER Program has made a number of different efforts aimed at facilitating the use of its advances by commercial entities. All TIPSTER materials, including research papers, are published and researchers are encouraged to present their results at other open forums. Some commercial participants have found their way to MUC and TREC, where their research can

be benchmarked and shared with others. TIPSTER participants have been encouraged to commercialize their ideas, when feasible. A number of commercial spin-offs of TIPSTER technology are happening. The TIPSTER Program is keeping abreast of developments in standards such as Z39.50 and the Document Management Alliance.

However, these efforts alone will not result in the rapid development of inexpensive, robust, and well-supported commercial products which also meet the Government analyst's requirements for information handling of text. Developing a product for commercial market itself takes time, so that advances promoted by TIPSTER, left to their own, might take longer than we wish to reach the commercial market. Commercial uses of the technology will not necessarily include all the functionality that Government analysts require. In order to get those more advanced versions of the technology for its use, the Government has to help push the final steps of technology transfer of those advanced features. This push can be accomplished during the development of a specific application, but these intermediate steps (generally steps 2-4) must be incorporated into the planning and budgeting for such an application.

The bottom line is that Government analysts, particularly Intelligence analysts, have text handling and information handling needs that are more difficult to satisfy than those of the general software buying and using commercial world. While the advanced software features they need may eventually become commonplace in commercial software, this is likely to take a long time to happen, since the general demand for such features appears to be moderate, at best. This is easy enough to understand if one stops to think about the kinds of applications that most potential users, of Detection for example, have. The general user - students, small businesses, people at home - certainly have very little interest in Recall and even in Precision. Their most pressing concern will be speed and getting one or two good answers to their questions in the top of their return document list. General business users, doing market research perhaps, or tracking competitors' activities, would likely be most interested in Precision in the top of their return document list; they are not tracking events at a level that requires a total and detailed picture of everything that has been said related to a particular topic over a long period of time. Additionally, the cost of missing something is measured in dollars, not the loss of life or the security of the country as it may be for an Intelligence Analyst. Even the applications, such as Insurance and Law, which have many similarities to the Intelligence application, rarely have the same far-reaching cost associated with failure as the Intelligence application. In addition, while these two applications

require better Recall than other commercial ones, they do not probably place the same stresses on a Detection tool because the user is searching in the context of a narrower range of document types and a narrower range of types of questions which they need to answer.

The need for Information Extraction technology appears even less pressing outside the Government. Besides the potential this technology may have to improve Detection capabilities, its major application to date is the filling of data bases from unformatted text to support further analytical tasks. While some non-Government applications, such as the development of formatted patient records from unformatted physician reports, have been investigated, there appears to be relatively low demand for this type of technology in the commercial world at this time.

Given this state of affairs, COTS products do not seem even yet to be an assured answer to the need for a well-supported suite of tools employing TIPSTER technology. Government off-the-shelf software, GOTS, may perhaps provide a better answer. Government owned software covers a broad spectrum of readiness and robustness. It offers no easy solution, because to provide software in a truly off-the-shelf condition to Government users, would in fact mean setting up a small business-like unit to do software testing and upgrades, promotion of products, distribution, integration support, and maintenance, all of which cost money over and above the initial investment in the software development. Thus, even though the Government owned software can be shared reasonably freely among agencies, costs related to the distribution would need to be born by someone, presumably the agency requiring the software. So the distribution of TIPSTER GOTS would require the establishment of a small business center to test and maintain the software. Before such a business could be established, a market survey would be required to determine if such a center would pay for itself, in addition to providing, at a comparable or lower price, better service than the already existing less formally organized system of distribution.

The system of distribution that exists now for TIPSTER GOTS is informal and low cost. We have a clearinghouse for information, in the form of the TIPSTER Executive Committee and other Government participants. The TIPSTER program freely advertises its involvement in Document Detection and Information Extraction through many informal contacts and a number of formal reviews and publications. Those desiring information about this type of software contact the committee or other participants and can be directed to contractors who have the kinds of software which is required. A Systems Engineering contractor also keeps on file records of the

design of all TIPSTER systems so that Government users can get detailed information about the configuration of any of the software which they may be interested in procuring. The cost for integrating, modifying, and maintaining the software is born by the using organization in fees to the vendor. So, TIPSTER GOTS is not free, but cheaper than developing the capability repeatedly in different locations.

The TIPSTER software Architecture was developed to permit such sharing of software developed at different agencies. However, it also is being promoted to meet a number of other problems associated with the delivery of software to the end user. The existence of an established set of interfaces for this group of technologies allows applications to be designed more quickly and with some accumulation of knowledge, across vendors, of the best ways to use them. It will allow the Government user to upgrade applications by the insertion of key new capabilities - for example, the replacement of one entity tagging module with a new one - without a complete change of application and without necessarily having to stay with the same vendor. The divorcing of the TIPSTER technologies from the human interface will make it easier to insert them seamlessly into a variety of user desktop environments. While all these improvements in the way the software is delivered and maintained will probably result in some cost savings, they should also make it easier and less disruptive for the user to upgrade systems, which is just as important.

The TIPSTER community began the development of the Architecture because of the need to accelerate the deployment of the technology it had developed. As with the technology itself, there did not appear to be sufficient interest in the commercial world to produce a set of agreed upon standards for integrating Document Detection and Information Extraction as quickly as the TIPSTER Program needed them. The Program, therefore, had to initiate this development itself. The Architecture has been developed to be as useful as possible to a variety of systems, since the TIPSTER community incorporates a widening sphere of vendors and researchers and requires a software Architecture at its core which allows all of them to work together without too great a cost in adaptation of individual systems.

## 6. Users

The development of new technology to support a user task must be a collaborative process. It is the user who knows what task has to be accomplished, but it is the technologist who understands what the

technology can do and how it can be configured to potentially help. The contributions of both groups are necessary. In the case of the TIPSTER Program most of the leadership has been taken by the technologists, albeit technologists with considerable experience in analytical tasks and knowledge of present and future user requirements. When the technologist is leading, an important part of any technology transfer project, we have found, is the establishment and maintenance of trust between the technologist and the user.

There are many facets to building and maintaining this trust. It requires understanding the user's style - what do they require from someone in order to trust them? Style will vary widely among different user organizations, from those who feel most comfortable with detailed project plans and schedule charts, to those who hate view graphs and only trust round table discussions with an occasional hand drawn diagram on the back of a sheet of paper. Some feel best with developers sitting in their midst, some like their developers working in hi-tech labs at the developers spaces. Some like formal reviews, others like people to stop by and chat, and so forth.

It is important for the technologist to communicate that he/she is promoting the user's ends, not the technologist's ends. The technology is being brought to serve the user's task and not solely to bring credit to the technology's developer. The technologist must understand the user's job well enough to explain clearly how the technology could possibly help. Demonstrations with the user's data, mocked up user interfaces, examples of similar projects, and descriptions of the results of testing the technology against tasks similar to the user's, can all be used to support the contention that the technology could be made helpful to the user. At the same time, the technologist, being an outsider, cannot prescribe how the task should be done with the new technology. It is the user's purview to control his/her own work style, after all. At the same time, if the technologist simply lets the user dictate how the job should be done, as often as not the user will make inappropriate use of the technology, either expecting too little or too much of its capabilities. The best solution seems to be for the technologist to offer as many reasonable choices as possible, for the application of the technology, allowing the user to choose or fine tune the actual work flow.

It is important also to educate the user about the technology. Better informed users will make better decisions about what technology to purchase and how to use it, so that the educational aspects of any work with the user will pay dividends far beyond the success or failure of any particular project. In the case of TIPSTER, the existence of an established and well



recognized program has been helpful in gaining user trust, but also the program workshops and evaluation conferences, as well as the proceedings coming out of them, have been useful in helping to explain to users what the technology does and how it works. The demonstration projects themselves were conceived of as partly having an educational mission; nothing would be so effective in persuading people that the technology "was for real," that it had something to offer them, and a bit about how it worked, than seeing it and trying it in an operational mode. If only one or two of these projects can succeed in making a large, positive contribution to a user's task, the returns should be substantial in making it clear to people that the technology is ready to be deployed and for what kinds of tasks.

The mission of educating the user about the technology means being as honest as possible about what it can and cannot do, explaining clearly how the technology will fit into the user's environment, discussing requirements on their time for development, evaluation, and training, and making good faith estimates of maintenance efforts, should a system become operational. In any project involving the joint investment of resources, a clear agreement worked out ahead of time concerning the obligations and resources coming from each party is important. Resources in this case means not only funding, but also the personnel time necessary to do things such as manage the project, develop requirements, review interface designs and so on. Generally, we have found that unless users are sufficiently concerned about the improvements in their task to invest time in learning about and supporting the application development, they are a "high risk" user, that is, one who is likely to back out before a project is completed or who will not use and support the software once it is completed. We have found that written agreements between technologists and customers, signed at an appropriate management level, even when operating within the same agency, covering expectations of funding and personnel resources as well as the criteria for success of a project, are extremely helpful in preventing disagreements and disappointments over the course of an applications development.

As with any job, the task of bringing an application to the operational environment will more likely be successful if the people who are involved work as a unified team. On the part of the project leader, frequently the TIPSTER representative, this requires including in the process from the beginning all those people who will be affected by the application. It means having representatives from the user group, from a couple levels of their management, from the developers and their management, from the IS (Information Services) staff or the equivalent,

cognizant of each other, of the goals and progress of the project, and aware of their respective roles and obligations. At the same time, the project leader must be aware that most of these people have other jobs to do and their time should be asked for only when necessary. This translates into carefully planned communication of information to those who need it, when they need it, well focused meetings, and the presentation to the user only of quality products (even interim products like prototype GUIs) which have been thoroughly reviewed and are free of obvious mistakes. To the extent that the project leader can also communicate the excitement of developing a new capability and create a team which enjoys working together, these factors will cause people to put forth their best effort to make the project successful.

## 7. Risks and Evaluations

In the end, however, it is well to remember that every project to transfer a new technology to the work place is a risk. While a great deal can and should be done to reduce the risk, there is never any guarantee that such a project will succeed in all or even most of its goals. As with any other endeavor, no failures means both that nothing will be learned and that nothing very difficult has been tried. So, some setbacks, at least, in transferring this technology to the user are necessary. It was problems, some of them not solved the first time, which taught us what I have recorded in this short discussion. At the same time, even an apparently failed project can have good results, if the causes of the failure are understood and the knowledge incorporated into later planning. For this reason, testing and evaluation of not only the research, but also applications is crucial.

Evaluation incorporates all aspects of the technology transfer project: (1) management, (2) relations with the user and IS staff, (3) development costs and schedule, (4) user interface, (5) work flow, (6) robustness of the software, (7) throughput, (8) accuracy, (9) ease of integration, (10) ease of maintenance, (11) use of the architecture. Evaluation methods differ for each. For project management issues (1,2,3), appropriate oversight by the management of the project leader is required; self evaluation at the end of the project, by both the project leader and the developer can be helpful. For usability issues (4,5) we have less experience. However, a current TIPSTER project is using a combination of interviews with user/testers after their sessions with the prototype and observations of the user/tester sessions to try to determine how easy to use the application is. Performance issues (6,7,8) can be evaluated by observation of the system for failures, for the speed of various tasks, and by comparison of

human generated output with machine output. Software design issues (9,10,11) are currently being evaluated through examination of the design in the TIPSTER Engineering Review Board, as well as by monitoring of the integration and maintenance phases by the project leader.

Much of the risk of a technology insertion project can be mitigated by two strategies - paying careful attention to the user issues addressed above and paying careful attention to the steps of technology transfer detailed above. Understanding how far along one's technology is on the path toward deployment and carefully evaluating the technology at each stage in its progress can prevent many mistakes. A number of difficulties I have observed in technology transfer projects have occurred because integration has occurred before component functions were sufficiently well researched and understood. Another common difficulty is failure to work out the best uses of the technology before an application is chosen and a user organization signed up. Both of these failings occur because needed steps in the progression of technology from research to the user environment have been short-changed.

A third strategy can also reduce the risk of technology transfer projects, which can be described as "start small and out of the way." This strategy does not reduce the chances of failure, but reduces the risk that failure will be disastrous. It has been important for TIPSTER to have demonstration projects which showed that TIPSTER technologies could do something useful and important. However, there are many important tasks that are at the same time "small," that is, well defined, and "out of the way," that is, limited in the number of users they affect. These have proven excellent opportunities to demonstrate the new technology without taking on all the additional problems of a very large and complicated development. These small-scale and limited applications have also simplified dealings with the user, since there were fewer of them immediately involved.

## 8. Conclusion

I have outlined in this paper much of what I have learned, from working in the TIPSTER Program, about transferring technology from the research stage to daily use in an operational environment. In working jointly on this endeavor we have all learned

these things and many of the ideas recorded here have been or are being incorporated into the Program to help guide future technology transfer. It is in the nature of a lessons learned, to support these future efforts, that I have recorded here, at the close of Phase II, so many of the difficulties that we have encountered and in many cases also overcome.

I have included in this paper many suggestions for dealing with the problems that arise in technology transfer. Technology transfer is a complex process and the number of issues that must be addressed is large. Thus, I hesitate to say that one set of issues is more important than any other. However, two of the issues I have discussed are of paramount importance. The first is making sure that the technology is ready to transfer - does it offer significant new capability and is it sufficiently well understood so that it can be made robust, fast, and be integrated with other necessary capabilities. The second is collaboration with the user. The technologist is neither a servant nor a savior, that is, neither the user nor the technologist has all the answers. What we have to do to transfer technology successfully, and what we have done in the TIPSTER Program, is make a place where the two can meet, user and technologist, to work collaboratively in a joint effort to tame the "text monster" and improve the way textual information is handled in our operational environment.

## 9. References

- [1] Sundheim, Beth; "*The Message Understanding Conferences*;" in this volume.
- [2] Harman, Donna K.; "*TREC Executive Summary*;" in this volume.
- [3] Pruett, Nancy J. and Tom Kinsella; "*Management of Free Text for NDIC: An Overview of the FTM Project*;" in this volume.
- [4] Sider, Ira; Jeffrey Baker, Deborah Brady, Lynne Higbie, and Tom Howard; "*Cable Abstracting and Indexing System (CANIS)*;" in this volume.
- [5] Kielty, John and Ira Sider; "*Advanced Data Extraction and Preparation Via TIPSTER (ADEPT)*;" in this volume.