Translingual Information Detection Extraction and Summarization

TEAM TIDES

October 2003

Testing Portability — Surprise Languages

Charles Wayne, cwayne@darpa.mil

Highlights:

- **Tech Note:** "Acquired 300,000 words of news translations in just 5 days by leveraging the contributions of more than 150 volunteers."
- **System Note:** "A significant advance in our ability to rapidly deliver multifaceted tools to analysts that require access to information in languages that they cannot read."

On the Inside:

Team TIDES is the newsletter of the TIDES program. The views expressed by the authors are their own, and do not necessarily reflect the views of their organizations or of DARPA. Comments and contributions should be directed to the new editor, Allison Powell.

TIDES is developing technology to enable English speakers to find and interpret needed information quickly and effectively regardless of language or medium. Most of the research is conducted on English, Chinese, and Arabic news data.

To see how quickly one could produce effective technology for unforeseen languages, we conducted two experiments where participants had no advance knowledge of what the languages would be. In March, we worked on Cebuano, the lingua franca in the southern Philippines, to find out how fast we could create or acquire essential linguistic resources. In June, we tackled Hindi to see how well key algorithms would work when linguistic resources are comparatively plentiful. Both succeeded.

The April issue of Team TIDES described the Cebuano experiment; this issue focuses on Hindi. Additional information will appear in the ACM Transactions on Asian Language Information Processing.

As indicated in the table, 15 groups participated actively in the Hindi experiment. We found that:

- Obtaining suitable linguistic resources – text, parallel text, translation lexicons, taggers, and morphological analyzers – is still the biggest challenge. The LDC played a crucial, central role; virtually everyone contributed; and distributed annotation worked. Converting Hindi to a common format required a lot of effort, but a shortage of relevant parallel English-Hindi translations was solved via a clever process described below by David Yarowsky.
- Translingual retrieval (finding Hindi documents from English queries) and tracking (finding Hindi documents from English examples) worked roughly as well for Hindi as for Chinese and Arabic.
- Automatic name tagging proved usable, but error rates were roughly double that of English.
- Generating English headlines for Hindi documents was moderately successful (67% of human performance compared to 77% for English documents).
- Readers could infer what translated documents were about, but not necessarily understand individual sentences. (Hindi-to-English reached 43% of human performance compared to 63% for Chinese-English and 80% for Arabic-English.)

There is no such thing as a "typical" language – each presents unique difficulties and surprises – but porting can be straightforward when there is adequate data. To cope with less resource-rich languages, one needs good techniques for obtaining data cheaply plus algorithms that work well with modest quantities. More work needs to be done to develop those methods and to calibrate performance. It could help to assemble resources in advance of a crisis.

The Hindi and Cebuano experiments both succeeded because groups collaborated beautifully – with extensive communication and common purpose. This would happen again in a real emergency.

	Alias-I	UC Berkeley	BBN	CMU	CUNY	Johns Hopkins	IBM	ISI	LDC	MITRE	NYU	SPAWAR	U. Sheffield	U. Massachusetts	U. Maryland
Resource Generation	■	■	■	■		■	■	■	■	■	■	■	■	■	■
Detection		■	■		■		■	■						■	■
Extraction	■		■			■				■	■	■	■		■
Summarization			■					■							■
Translation				■		■	■	■							

Coping With Surprise: Responsive Language Technology

Doug Oard, oard@umd.edu

Anticipating future hot spots has proven to be a challenge. Would Somalia have been on your list? Rwanda? Haiti? Albania? If it were your business to have your language technology ready for anything, then your list would include those places and hundreds more. And when a deployment order comes, time is a luxury we cannot afford; deployment timelines for urgent operations are measured in days or weeks.

This is not a good match to the way people used to develop language technology. At the risk of oversimplifying to make a point, the "old way" involved studying a language intensely and then hand-crafting systems that could automatically accomplish critical tasks: search, translation, summarization, and extraction. Timelines were typically measured in years, and achieving the full potential of the most challenging of these — translation — sometimes required decades. Rarely was it possible to clearly discern requirements over such time spans; as a result, the required investment (which could grow quite large over time) could only rationally be made in a few cases. The net effect was that we have sometimes found ourselves operating with less access to mission-critical information than we need.

For the past decade or so, a growing group of innovative researchers have been developing a "new way," one that promises a far better match between our needs and the capabilities of the systems that we build. Instead of people studying a language, we now have machines that study the way people use a language. Instead of hand-crafting our systems, we now build machines that mimic the way they see people using that language. Our machines still don't understand the meaning of the words that they produce. But that is not important, so long as the people that read those words are able to understand them. And, to an increasing degree, they can.

But there can be an enormous difference between describing a potential and demonstrating a capability. Machines cannot mimic the way people use a language unless we can show them examples. And not just a few examples; our present systems need to see millions of words. And not just any words; words that have been translated by people. And not just any translated words; words that are translated the way that you want the machine to translate them. It does little good to show a machine translated poetry if you will then ask it to translate tomorrow's news. Where do you find this "training data?" Ask TIDES.

In June, 2003, about a hundred of the world's top language technology researchers set out to demonstrate what could be done. DARPA chose Hindi, a language spoken by over 400 million people. At that time, there were no broad-coverage translation systems from Hindi into English, and almost no Hindi search capability in any of the world's major Internet search engines. It takes a whole day to fly from Washington to India; in that time the first translation system was built and the first search capabilities were created. It can take a month to load a ship and sail it from the United States to the Indian Ocean; over that period, our systems matured to provide a broad array of useful capabilities. The articles in this issue of Team TIDES have been selected to tell at least part of the story of how that was done. In the history of every innovation, there is one moment that separates the past from the future, the moment when potential becomes reality. For rapidly responsive language technology, that moment was in June of 2003.

Hindi 101

Sanjeev Khudanpur (khudanpur@jhu.edu)

Hindi is the national language of India, with roughly 300 million native speakers. Another 100 million or more use Hindi as a second language. It is the language of dozens of major newspapers, magazines, radio and television stations, and of other media. It is the *lingua franca* of the Indian armed forces. It is closely related to many other languages spoken on the Indian subcontinent and serves as a *bridge language* for technology development in Bengali, Punjabi, Gujarati, Marathi and Urdu. For instance, technology for automatically extracting names of people mentioned in a document, once developed for Hindi, can be quickly ported to the other languages such as Bengali.

Hindi grammar and morphology are rooted in Sanskrit. Nouns and pronouns are inflected to mark gender, number and case. Adjectives are inflected to agree with their nouns, and verbs have a fairly complete morphological paradigm to indicate number, gender, tense, etc.

Hindi is written left-to-right in the Devanagari script, with space-separated words. Characters within words are joined by an over-line, for example जॉर्ज बुश (George Bush). Unlike the concatenative nature of English letters, however, Devanagari is a rule-based *syllabary*. Put differently, English words are written by concatenating individual characters, while Hindi words are written by concatenating individual syllables. For example, the first syllable "i" in the word *Iraq* is written as इ. But the first syllable "ri" of the word *risen* is written as रि, bearing no visual resemblance to its constituent vowel "i". There are tens of thousands of syllables in Hindi, but a Hindi writer does not have to memorize how to write each syllable. Each Hindi syllable has a regular [C]*V internal structure, with zero-or-more consonants and a vowel. There are 33 consonants and 12 vowels, and there are linguistic rules which govern how a combination of consonants and vowel is written.

Computer-internal representations of Devanagari (e.g. Unicode) use multiple bytes or "code-points" to store each syllable, with ad hoc rules for composing and rendering these code-points for display and printing. This opportunity for ad hoc choices has led to the proliferation of dozens of competing encoding standards, and it is annoyingly routine to find electronic texts in one encoding which are unreadable in another. This causes a key bottleneck in accessing information in

Hindi texts, and is a major reason why commercial search engines do not address Hindi. Manual effort by a native speaker familiar with character-encoding issues was needed to develop programs for converting each such encoding to a common standard (UTF-8) for use in the TIDES Surprise Language experiments. This is a valuable lesson for other programs interested in rapidly producing language technologies because the problem is likely to resurface in many other languages of interest.

Resource Generation for the Surprise Language Exercise

Mike Maxwell (maxwell@ldc.upenn.edu)

The Surprise Language exercise started on June 2, but preparation began at the Linguistic Data Consortium (LDC) months before, when we embarked on a survey to identify candidate languages. By June we had surveyed approximately 150 languages, identifying resources such as news text, bilingual text, electronic bilingual lexicons and morphological parsers.

The announcement of the Surprise Language meant putting the survey effort into high gear for the chosen language—Hindi. It was time to be sure we had all the resources the other TIDES sites would need, and that the resources were of sufficient quality. Quantity of Hindi resources was not an issue. Our goal had been to find 100K words of news text. The pre-exercise survey had uncovered much more than this; and a day or two into the exercise, we had downloaded three orders of magnitude more text than the target amount.

One issue that became obvious early on was that while there were two standard encoding systems for Hindi (Unicode and ISCII, an 8-bit encoding developed in India), neither encoding was in significant use, at least inside India. Instead, every web-based news site had at least one proprietary font, and each font used a proprietary encoding. We therefore had to find or build programs to convert from non-standard encodings into one of our standards. This was not a simple matter of mapping one code point to another. There were more symbols in the Devanagari writing system than there were code points in a 7- or 8-bit encoding and the choice of glyphs varied across fonts and encodings. Converters tended to handle core cases well, but failed to produce clean output on large texts. A good deal of effort was spent on building and debugging these converters, with problems surfacing even on the last day of resource collection.

We found another surprise half way into the exercise: a lack of parallel bilingual news text. We had found several bilingual news sites on the first day, but it turned out that the Hindi and English news stories were not translations but were instead written independently. Virtually none of the news sites contained parallel Hindi-English text. Fortunately, we had large quantities of parallel text in other genera, and eventually found a small amount of parallel news text.

A parallel effort to resource collection involved resource creation: annotating text, particularly tagging named entities. While we found some web resources to help with this process (such as gazetteers), most of our effort involved recruiting and training Hindi speakers to do annotation manually.

The LDC was of course not the only site involved in resource collection and creation; other sites that participated in resource generation are listed in the table on page one. An email list was set up for the exercise, and frequent teleconferences helped coordinate what was truly a team effort.

In the end, we were able to meet all our goals, as shown in the following table. Most importantly, we demonstrated that the TIDES team can build a language capability where none existed, and on short notice!

Resource	Goal	Result (approximate)
Monolingual News Text	100K words	100M words
Aligned Bilingual Text	100K words	2M words (incl. 400K words of news text)
Electronic Bilingual Lexicon	10K words	30K words
Named Entity Tagged Text	100K words	425K words
Morphological Parser	Yes	Yes

Technical Note:

Scalable Elicitation of Training Data for Machine Translation

David Yarowsky, yarowsky@jhu.edu

The tremendous advances in TIDES machine translation quality over the past few years depend on one key resource: a large collection of representative translation-equivalent “parallel” text. With enough time, the Linguistic Data Consortium can assemble enormous collections of this crucial training data, and the resulting systems for Arabic and Chinese are now among the best in the world. Time is, however, the one thing we did not have for Hindi. Although the TIDES team was able to harvest a substantial amount of existing parallel text in a month, over one million words, the overwhelming majority of this shared data was not from news sources. Since our goal was to translate news stories, this was a critical weakness.

Hindi is widely spoken, so TIDES teams at CMU, USC-ISI, and the University of Maryland hired people locally to translate news stories. Over a three-week period, translations for about 30,000 words of news were obtained at a cost of about ten cents per word, somewhat below what professional translation bureaus would charge. These translations played a key

role in system development across the program, but much more would be needed to realize the full potential of today's statistical machine translation techniques. Johns Hopkins University therefore set out to develop a **translation elicitation server** that ultimately acquired 300,000 words of news translations for less than two cents per word in just 5 days (including all administrative costs) by leveraging the contributions of more than 150 volunteers from around the planet.

Inspired by the Open Mind Initiative for the collaborative collection of human knowledge, volunteers were able to participate using a simple Web interface. We incentivized participation by offering electronic Amazon.com gift certificates as prizes for the best translations. These rewards were sent to the winners by email each night, rapidly building trust and providing near-immediate feedback. Automatic evaluation of translation quality, using the BLEU metric developed by IBM, was the key to making this work. We began with a set of reference translation pairs from the initial set produced at USC-ISI and Maryland. One out of every five sentences translated by a volunteer was chosen from this set and used to compute the BLEU score. Sentences from the best volunteer translators (as ranked by BLEU) were incrementally added to the pool of test sentences, thus making the design infinitely scalable. An important byproduct of this approach was multiple Hindi translations for a large set of English sentences, a unique resource for research on the analysis and generation of paraphrases.

Starting with just a few newsgroup postings and emails to Hindi-speaking acquaintances, recruitment propagated by word of mouth. Participation increased exponentially during the 5-day experiment, with volume doubling roughly every 1.5 days. All four groups that built machine translation systems used the resulting translations, and contrastive experiments showed improvements in translation quality as a result. The framework offers great promise for rapidly harnessing a worldwide talent pool in any case where tailored training data is needed and automated evaluation metrics are available.

System Note:

CuSTaRD: Integrating Language Technologies for Hindi

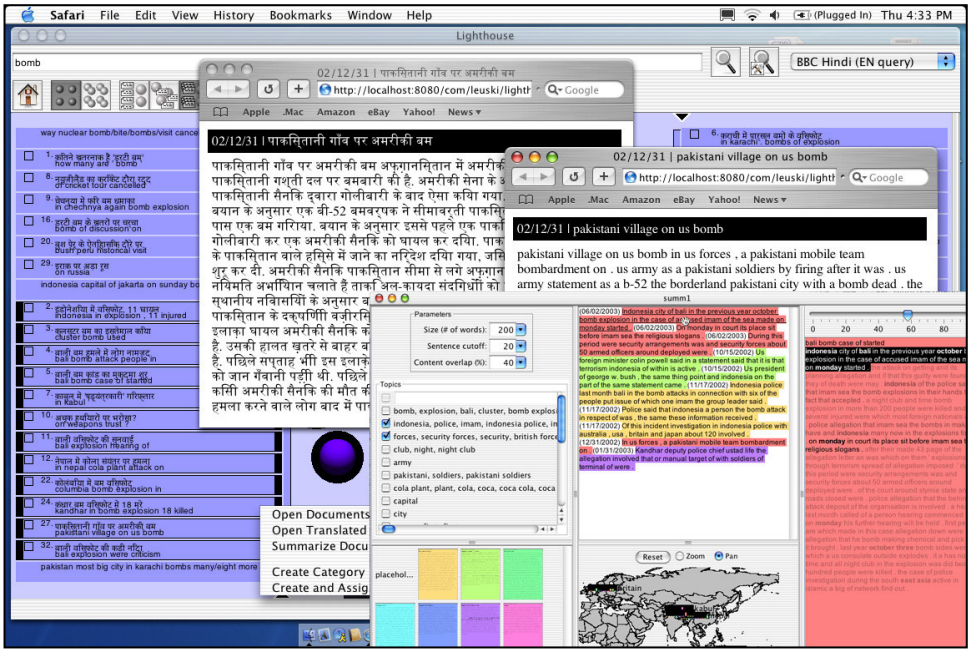
Eduard Hovy, Anton Leuski and Chin-Yew Lin (hovy,leuski,cyl)@isi.edu

Helping analysts find essential information in languages they cannot read calls for integrated systems that offer multiple ways of helping people understand what they are looking at. We have built an interactive information analysis environment that we call CuSTaRD (Clustering, Summarization, Translation, Reformatting and Display) from four major components: (1) cross-language information retrieval, clustering, and visualization; (2) multi-document headline generation; (3) rich multi-document summarization; and (4) automatic language translation. The screen shot on the right illustrates some of the ways in which these capabilities are presented to the analyst.

Searches in CuSTaRD begin with a query that can be posed in either English or Hindi. English queries are translated into Hindi using techniques from the University of Maryland, and the resulting query is presently used to search a small demonstration collection that contains about 3,000 Hindi news stories. Users can select a subset of the highly ranked documents to manipulate in different views or to cluster and summarize. Each Hindi document is represented by a Hindi headline and an automatically created English translation of that headline.

Analysts can summarize selected clusters using just-in-time multi-document summarization. The system depicts relationships between summaries, documents, and topic clusters. English headlines are created automatically for each cluster to give the analyst a quick overview of cluster content. Named entity tagging (also known as named entity recognition) from BBN can also be used as a basis for automatically creating alternative map, timeline, and entity-relationship visualizations if desired.

We designed our system architecture for easy extensibility in order to rapidly integrate these diverse components. All of the Hindi documents were translated into English when they were first indexed, thereby simplifying the integration of



components that had been designed originally for English. This proved to be a good decision; automatic generation of headlines from the English translations yielded far better headlines (59% better by one measure) than first generating headlines in Hindi then translating them to English. Of course, on-demand translation will ultimately be needed for scalable applications. This should be practical with only modest improvements in the latency of our translation engine that are now well within the present state of the art.

CuSTaRD represents a significant advance in our ability to rapidly deliver multifaceted tools to analysts that require access to information in languages that they cannot read. The degree of integration that we have achieved also serves as a source of inspiration for our future research, helping us to see new opportunities that previously were at the intersection of separately developed capabilities.

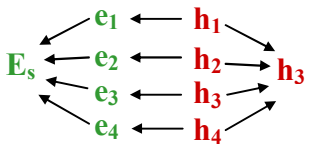
Searching Hindi using Statistical Stemming

Fredric Gey, gey@ucdata.berkeley.edu and Aitao Chen, aitao@sims.berkeley.edu

Cross-language information retrieval poses two key challenges: identifying meaningful Hindi terms that characterize the content of a document, and relating those terms to the English terms that an analyst might choose to express their query.

For many languages, including Hindi, words can easily be split at white space. But words are often not the best basis for search; in English, a search for "attacked" often should retrieve documents that report on "attacks." Stemming both to "attack" by stripping suffixes is a common way to accommodate such variations.

The first available Hindi stemmer, built and tested by another TIDES team from a published technique, yielded disappointing results. Such problems are not unusual; a stemmer must strike a balance between precise matches and comprehensive coverage, and that can require extensive tuning. Excellent stemmers are available for English, of course, so we tried training a stemmer for Hindi here at Berkeley using an English stemmer and a set of known translation relationships between English and Hindi. The key idea is simple: words that express the same meaning should be grouped together, as illustrated by the following sketch:



Here, **{h_1, h_2, h_3, h_4}** are Hindi words and **{e_1, e_2, e_3, e_4}** are corresponding English translations that stem to $\mathbf{E_s}$. We simply choose one Hindi word (h_3 in this example) to represent every word in the set. We refer to this as *statistical stemming* because we don't really know the translation relationships; rather, we infer them using statistics from a TIDES machine translation system. Using translations from USC-ISI, our statistical stemmer grouped 89,650 unique Hindi words into 26,436 clusters, for an average of 3.4 words for each "stem." Our initial experiments using a small test collection developed jointly by BBN and the University of Maryland showed an 11% relative increase in search effectiveness from our stemmer, and the final official evaluation on 41,697 Hindi documents showed somewhat smaller improvements (between 3.1% and 7.2%).

Berkeley chose automatic query translation to address the second key challenge, crossing the language barrier. We used the small BBN/Maryland test collection for refinement of techniques, ultimately combining four translation lexicons (from IBM, NYU, USC-ISI, and Berkeley) and bilingual lists of country names and the names of 250 Indian cities and states that we assembled at Berkeley. The efficiency of our system was also improved using a "stopword" list from the University of Massachusetts that identified low-content terms that could safely be ignored.

Library catalogs contain book titles in many languages; we found 44,366 Hindi catalog records in the University of California library and another 28,601 records from a library in the United Kingdom. From these we built statistical mappings between 10,839 unique words from the English subject heading in each record and the Hindi words in the associated book title. Example associations include "police" (pulisa), "Pakistan" (Pakistana) and "terrorism" (atankavada). These Hindi words were represented in a standard transliteration developed by the Library of Congress, but conversion to our standard UTF-8 encoding did not become available in time to allow use of this data in our search system.

Berkeley's final results were quite good, yielding more than three relevant documents among the top five on average. We relied heavily on resources created across the program, so our success is due in large measure to the outstanding cooperation in the TIDES team.

Rapid Development of Cross-Lingual Question Answering using Information Extraction

Satoshi Sekine, sekine@cs.nyu.edu

Name identification and classification is an important aspect of information extraction and a key component in many applications. For the surprise language exercise, our team at New York University demonstrated our newly developed Hindi named entity tagger in what we believe is the first English-Hindi cross-lingual question answering system. Our tagger recognized three types of names (person, organization and location) and ten types of numeric expressions (e.g., dates and monetary

amounts). The resulting question answering system found correct answers for questions such as “Who is the prime minister of India?”, “When did India explode an atomic bomb?”, “How much money was involved in the Bali bomb attack?” and “What is the length of the planned gas pipeline to be brought from Iran to India via Pakistan?” The system receives a question in English, finds possible answers in a half a year of Hindi news stories, and displays the answer candidates in English. Users interact with the system over the Web.

The main components of the system were developed for TIDES: our named entity tagger for information extraction, cross-language information retrieval ideas from CMU, and an on-demand machine translation service provided by USC-ISI.

To this we added an English question analyzer that we had previously developed and an answer finder for Hindi that we developed specifically for this purpose. Although we created our system within a month, this would not have been possible without the TIDES team. Most of the data was provided by other groups, and many ideas incorporated into our system were suggested on the surprise language mailing list. The documents we searched were provided by MITRE; we obtained lexicons from seven teams (LDC, IBM, CMU, SPAWAR, University of Sheffield, USC-ISI, and BBN) and name-annotated text from BBN and LDC, and we obtained other tools from the University of Maryland and Alias-i.

In practice, more time was spent on preparing and massaging the data than on the actual creation of the system. Character encoding was one of the major problems, but properly using the data and merging different data sources are also troublesome when native speakers are not available. Although we did not have much time to tune the system, it was nevertheless able to place at least one correct answer somewhere in the top 10 candidates for about half (27 of 56) of the questions in a small test collection that we built for this purpose. In short, we created a cross-lingual question answering system in one month, for which the data and tools provided by the TIDES team were essential. So this was like a 6 month project rolled up into one month. I enjoyed the feeling of working together in this manner. Thanks for this opportunity!

Why This Works: Incorporating New Languages in Days, Not Years

Mark Liberman, myl@ldc.upenn.edu

For 29 days in June, TIDES researchers worked to create systems for translating from Hindi to English. Finding a point of reference for their accomplishments can be challenging; there simply were no comparable systems for this task before the TIDES effort started. The quality of the "instant Hindi machine translation" was amazing, given the incredibly short development cycle.

Maybe the potential of this technology should have been obvious. Anyone who has been following developments in machine translation knows that the methods used by TIDES researchers are mostly language-independent, with most of the language-dependent parts "learned" automatically from text corpora. In fact, rebuilding the TIDES systems for Hindi generally took just a few days. Most of the month was spent preparing the "raw materials"–especially translation-equivalent “parallel” text–that provided the basis for automatic training. The same proved to be true for the other human language technologies that were tested on Hindi. The quality of systems available for a given language depends on the quantity and suitability of the training data available for that language much more than it depends on the amount of human engineering put into it. The technologies still need improvement, of course, but we have certainly shown that present capabilities can be rapidly obtained in new languages if the supporting data is there.

Give these TIDES engineers some data to stand on, and they can move the linguistic world. Perhaps it's time for a systematic effort to supply this foundation for the few hundred languages that cover nearly all of the world's speaking, writing and reading.

Editors:
Doug Oard (past editor) oard@umd.edu
Allison Powell apowell@cnri.reston.va.us
Layout:
Erika Barragan-Nunez (USC-ISI)
Anton Leuski