Basic Language Resources for Diverse Asian Languages: A Streamlined Approach for Resource Creation

Heather Simpson, Kazuaki Maeda, Christopher Cieri
Linguistic Data Consortium
University of Pennsylvania
3600 Market St., Suite 810
Philadelphia, PA 19104, USA
{hsimpson, maeda, ccieri}@ldc.upenn.edu

Abstract

The REFLEX-LCTL (Research on English and Foreign Language Exploitation-Less Commonly Taught Languages) program, sponsored by the United States government, was an effort in simultaneous creation of basic language resources and technologies for under-resourced languages, with the aim to enrich sparse areas in language technology resources and encourage new research. We were tasked to produce basic language resources for 8 Asian languages: Bengali, Pashto, Punjabi, Tamil, Tagalog, Thai, Urdu and Uzbek, and 5 languages from Europe and Africa, and distribute them to research and development also funded by the program. This paper will discuss the streamlined approach to language resource development we designed to support simultaneous creation of multiple resources for multiple languages.

1 Introduction

Over the past decade, the scope of interest in language resource creation has increased across multiple disciplines. The differing approaches of these disciplines are reflected in the terms used for newly targeted groups of languages. Less commonly targeted languages (LCTLs) research may focus on development of basic linguistic technologies or language-aware applications. The REFLEX-LCTL (Research on English and Foreign Language Exploitation-Less Commonly Taught Languages) program, sponsored by the United States government, was an effort in simultaneous creation of basic language resources and technologies for LCTLs, namely languages which have large numbers of speakers, but are nonetheless infrequently studied by language learners or researchers in the U.S.

Under the REFLEX-LCTL program, we produced "language packs" of basic language resources for 13 such languages. This paper focuses on the resources created for the 8 Asian languages: Bengali, Pashto, Punjabi, Tamil, Tagalog, Thai, Urdu and Uzbek.[1]

Our approach to language pack creation was to maximize our resources. We accomplished this in a number of ways. We engaged in thorough planning to identify required tasks and their interdependencies, and the human and technical resources needed to accomplish them. We focused intensely on identifying available digital resources at the start of language pack creation, so we could immediately begin assessment of their usability, and work on filling the resource gaps identified. We developed annotation tools usable for multiple tasks and all languages and designed to make manual annotation more efficient. We developed standards for data representations, to support efficient creation, processing, and end use.

2 Planning For Basic Language Resources Creation

2.1 Language Selection

The selection of REFLEX-LCTL target languages was based on a number of criteria, while operating within a fixed budget. The most basic criterion was that the language be spoken by large number of native speakers, but poorly represented in terms of available language resources. The Indic languages (Bengali, Punjabi, Urdu) were chosen to give researchers the opportunity to experiment with bootstrapping techniques with resources in related languages. In order to maximize the usefulness and generality of our methods, the project adopted the additional goals of variation in the expected availability of raw resources and also variation in linguistic characteristics both within the set of selected languages and in comparison to more well-resourced languages. Though we are focusing, in this paper, on the Asian languages, LCTL languages are linguistically and geographically diverse, representing major language families and major geographical regions.

The following short descriptions of the Asian LCTL languages are intended to provide some context for the language pack discussions. The lan-

[1]The other languages were: Amazigh (Berber), Hungarian, Kurdish, Tigrinya and Yoruba.

guage demographic information is taken from Ethnologue (Gordon, 2005).

2.2 Bengali

Bengali is spoken mostly in Bangladesh and India. The language pack for Bengali was the first complete language pack we created, and it served as a pilot language pack for the rest of the project. There were a relatively large number of raw materials and existing lexicons to support our lexicon development, and our language pack included the largest lexicon among the Asian language packs.

2.3 Urdu

Urdu, spoken primarily in Pakistan and part of India, is closely related to Hindi. Urdu is traditionally written using Perso-Arabic script, and has vocabulary borrowed from Arabic and Persian. This was a language which had a large amount of available digital resources in comparison with other LCTLs, but did not meet our original expectations for raw digital text.

2.4 Thai

Thai is a Tai-Kadai language spoken by more than 20 million people, mainly in Thailand. Thai was another language which was relatively rich in available digital language resources. The Thai language pack includes the largest amount of monolingual text and found parallel text among the language packs. For Thai, tokenization, or word segmentation, was probably the most challenging aspect of the resource creation effort. For the initial version of the language pack, we used a tokenization tool adopted from an opensource software package.

2.5 Tamil

Tamil is a Dravidian language with more than 60 million speakers in India, Sri Lanka, Singapore and other countries. We benefited from having local language experts available for consultation on this language pack

2.6 Punjabi

Punjabi, also written as Panjabi, is an Indo-European language spoken in both India and Pakistan. Ethnologue and ISO 639-3 distinguish three variations of Punjabi: Eastern, Mirpur and Western, and the Eastern variation has the largest population of speakers. We were able to obtain relatively large amounts of monolingual text and existing parallel text.

2.7 Tagalog

Tagalog is an Austronesian language spoken by 15 million people, primarily in the Philippines. The monolingual text we produced is the smallest (774 K words) among the eight Asian language packs, due in part to the prevalence of English in formal communication mediums such as news publications.

2.8 Pashto

Pashto an Indo-European language spoken primarily in Afghanistan and parts of Pakistan. It is one of the two official languages in Afghanistan. Ethnologue and ISO 639-3 distinguish three varieties of Pashto: Northern, Central and Southern. Major sources of data for this language pack included BBC, Radio Free America and Voice of America.

2.9 Uzbek

Uzbek is primarily spoken in Uzbekistan and in other Asian republics of the former Soviet Union. The creation of the language pack for Uzbek, which is a Turkic language, and the official language of Uzbekistan, was outsourced to BUTE (Budapest University of Technology and Economics) in Hungary. Even though the Uzbek government officially decided to use a Latin script in 1992, the Cyrillic alphabet used between the 1940's and 1990's are still commonly found. Our language pack contains all resources in Latin script and includes an encoding converter for converting between the Latin script and the Cyrillic script.

2.10 Designing Language Packs

Within the REFLEX-LCTL program, a language pack is a standardized package containing the following language resources:

- Documentation
  - Grammatical Sketch
- Data
  - Monolingual Text
  - Parallel Text
  - Bilingual Lexicon
  - Named Entity Annotated Text
  - POS Tagged Text
- Tools
  - Tokenizer
  - Sentence Segmenter
  - Character Encoding Conversion Tool
  - Name Transliterators
  - Named Entity Tagger
  - POS Tagger
  - Morphological Analyzer

Grammatical sketches are summaries (approximately 50 pages) of the features of the written language. The primary target audience are language engineers with a basic grounding in linguistic concepts.

Monolingual text is the foundation for all other language pack resources. We provided monolingual text in both tokenized and non-tokenized format. Parallel text is an important resource for development of machine translation technologies, and allows inductive lexicon creation. The bilingual lexicons also support a variety of language technologies. The named entity annotations and part of speech tagged text can be used to create automatic taggers.

The language packs also include basic data processing and conversion tools, such as tokenizers, sentence segmenters, character encoding converters and name transliterators, as well as more advanced tools, such as POS taggers, named entity taggers, and morphological analyzers.

These language packs include 6 of the 9 text resources and tools in 4 of the 15 text-based modules listed in the current BLARK matrix (ELDA, 2008).

When we had a relatively stable definition of the deliverables for language pack, we were able to begin planning for the downstream processes

3 Standards for Data Representation

An important step in planning was to define standards for language pack data representation, to allow all downstream processes to run more efficiently.

3.1 Language Codes

We decided to use the ISO 639-3[2] (also Ethnologue, 15th edition (Gordon, 2005)) three-letter language codes throughout the language packs. For example, the language code is stored in every text data file in the language packs. The ISO 639-3 language codes for our eight languages are as follows: Urdu (URD), Thai (THA), Bengali (BEN), Tamil (TAM), Punjabi (PAN), Tagalog (TGL), Pashto (PBU) and Uzbek (UZN). When there were multiple ISO 639-3 codes for a target language, the code for the sublanguage for which the majority of the written text can be obtained was used.

3.2 File Formats

One of the first tasks in planning for this data creation effort was to define file formats for the monolingual text, parallel text, lexicons and annotation files. This designing process was led by us and a group of experts selected from the research sites participating in the REFLEX-LCTL program. The requirements included the following:

- Monolingual and parallel text files should be able to represent sentence segmentation, and both tokenized and non-tokenized text.
- For parallel text, the text and the target language and the translations in English should be stored in separate aligned files.
- Unique IDs should be assigned to sentences/segments, so that the segment-level mapping in parallel text is clear.
- Annotation files should be in a *stand-off* format: i.e., annotations should be separate from the source text files.
- Lexicon files should be able to represent at the minimum of word, stem, part-of-speech, gloss and morphological analysis.
- File formats should be XML-based.
- Files should be encoded in UTF-8 (UNICODE).

After several cycles of prototyping and exchanging feedback, we settled on the following original file formats named "LCTL text format" (LTF - file name extension: .ltf.xml), "LCTL annotation format" (LAF - file name extension: .laf.xml), and "LCTL lexicon format" (LLF - file name extension .llf.xml. Appendix A shows the DTD for LTF format.

3.3 Evaluation and Training Data Partition

To support evaluation of language technologies based on the data included in the language packs, we designated approximately 10% of our primary data as the evaluation data and the rest as the training data. Any data that was included as "as-is", (e.g. found parallel text), was included in the training partition.

3.4 Directory Structure and File Naming Conventions

Giving all language packs a consistent design and structure allows users to navigate the contents with ease. As such, we defined the directory structure within each language pack to be the following.

The top directory was named as follows.

```
LCTL_{Language}_{Version}/
```

For example, the version 1.0 of the Urdu language pack would have the top directory named LCTL_Urdu_v1.0.

The top directory name is also used as the official title for the package, so the full name rather than the language code was used for maximum clarity for users not familiar with the ISO coding.

Under the main directory, the following subdirectories are defined:

[2]See `http://www.sil.org/iso639-3/` for more details

```
Documentation/
Grammatical_Sketch/
Tools/
Lexicon/
Monolingual_Text/
Parallel_Text/
Named_Entity_Annotations/
POS_Tagged_Text/
```

The Parallel_Text directory was divided into "Found" and "Translation" directories. The Found directory contains parallel text that was available as raw digital text, which we processed into our standardized formats. The Translation directory contains manually translated text, created by our translators or subcontractors as well as part of found parallel text which we were able to align at the sentence-level. The data directories (e.g., Monolingual Text, Parallel Text, Named Entity Annotation, POS Tagged Text) were further divided into evaluation data ("Eval") and training data ("Train") directories as requested by the program.

We used the following format for text corpora file names wherever possible:

```
{SRC}_{LNG}_{YYYYMMDD}.{SeqID}
```

{SRC} is a code assigned for the source; {LNG} is the ISO 639-3 language code; {YYYYMMDD} is the publication/posting date of the document; and {SeqID} is a unique ID within the documents from the same publication/posting date.

4 Building Technical Infrastructure

In creating the language resources included in the LCTL language packs, we developed a variety of software tools designed for humans, including native speaker informants without technical expertise, to provide data needed for the resource creation efforts as efficiently as possible. In particular, the following tools played crucial roles in the creation of language packs.

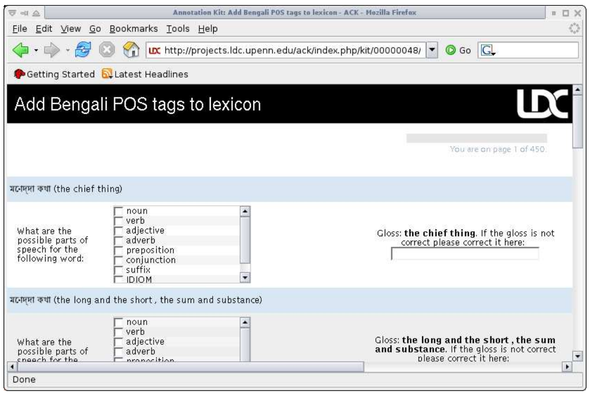
4.1 Annotation Collection Kit Interface (ACK)

In order to facilitate efficient annotation of a variety of tasks and materials, we created a web-based judgment/annotation tool, named the Annotation Collection Kit interface (ACK). ACK allows a task manager to create annotation "kits," which consist of question text and predefined list and/or free-form answer categories. Any UTF-8 text may be specified for questions or answers. ACK is ideal for remote creation of multiple types of text-based annotation, by allowing individual "kits" to be uploaded onto a specific server URL which any remote user can access. In fact, using this tool we were able to support native speaker annotators working on part-of-speech (POS) annotation in Thailand.

When annotators make judgments in ACK, they are stored in a relational database. The results can be downloaded in CSV (comma-separated value) or XML format, so anyone with secure access to the server can easily access the results.

ACK was designed so that anyone with even a basic knowledge of a scripting language such as Perl or Python would be able to create the ACK annotation kits, which are essentially sets of data corresponding to a sets of annotation decisions in the form of radio buttons, check boxes, pull-down menus, or comment fields. Indeed some of the linguists on the LCTL project created their own ACK kits when needed. Although they are limited in scope, creative use of ACK kits can yield a great deal of helpful types of annotation.

For example, for POS annotation, the annotators were given monolingual text from our corpus, word by word, in order, and asked to select the correct part of speech for that word in context. We also used ACK to add/edit glosses and part of speech tags for lexicon entries, to perform morphological tagging, and various other tasks that required judgment from native speakers.


Annotation Kit: Add Bengali POS tags to lexicon - ACK - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://projects.ldc.upenn.edu/ack/index.php/kit/00000048/
Go
Getting Started
Latest Headlines
LDC
Done

Add Bengali POS tags to lexicon

You are on page 1 of 450.

মোদ্দা কথা (the chief thing)

What are the possible parts of speech for the following word:

- noun
- verb
- adjective
- adverb
- preposition
- conjunction
- suffix
- IDIOM

Gloss: **the chief thing**. If the gloss is not correct please correct it here:

মোদ্দা কথা (the long and the short , the sum and substance)

What are the possible parts of speech for the

- noun
- verb
- adjective
- adverb
- preposition

Gloss: **the long and the short , the sum and substance**. If the gloss is not correct please correct it here:

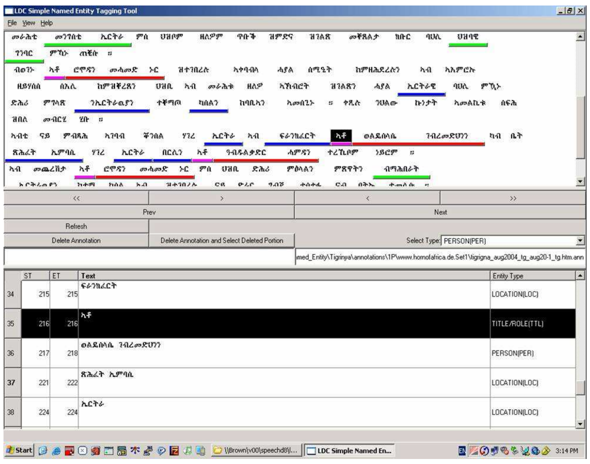
Figure 1: ACK - Annotation Collection Kit

Figure 1 shows a screen shot of ACK.

4.2 Named Entity Annotation Tool

For named entity annotation task, we chose to employ very simple annotation guidelines, to facilitate maximum efficiency and accuracy from native-speaker annotators regardless of linguistic training.

We used an named entity (NE) annotation tool called SimpleNET, which we previously developed for the named entity annotation task for another project. SimpleNET requires almost no training in tool usage, and annotations can be made either with the keyboard or the mouse. The NE annotated text in the LCTL language packs was created with this tool. This tool is written in Python using the QT GUI toolkit, which allows the display of bidirectional text.


LDC Simple Named Entity Tagging Tool
File View Help
<<
>
<
>>
Prev
Next
Refresh
Delete Annotation
Delete Annotation and Select Deleted Portion

Select Type: PERSON(PER)

med_Entity\Tigrinya\annotations\1P\www.homofafrica.de.Set1\tigrigna_aug2004_tg_aug20-1_tg.htm.ann

	ST	ET	Text	Entity Type
34	215	215	ፍራንክፈርት	LOCATION(LOC)
35	216	216	ኣቶ	TITLE/ROLE(TTL)
36	217	218	ወልደስላሴ ገብረመድህን	PERSON(PER)
37	221	222	ጽሕፈት ኢምባሲ	LOCATION(LOC)
38	224	224	ኤርትራ	LOCATION(LOC)

Figure 2: SimpleNET Annotation Tool

Figure 2 shows a screen shot of SimpleNET.

4.3 Named Entity Taggers and POS Taggers

We created common development frameworks for creating named entity taggers and part-of-speech taggers for the LCTL languages. These frameworks allowed us to create taggers for any new language given enough properly-formatted training data and test data. Included are core code written in Java as well as data processing utilities written in Perl and Python. The framework for creating POS taggers was centered around the MALLET toolkit (McCallum, 2002).[3]

4.4 Data Package Creation and Distribution

As per LDC's usual mechanisms for small corpora, language packs were to be packaged as a tar gzip (.tgz) file, and distributed to the REFLEX-LCTL participating research sites. The distribution of the completed languages packs were handled by our secure web downloading system. Access instructions were sent to the participating research sites, and all downloads were logged for future reference.

5 Steps for Creating Each Language Pack

5.1 Identifying Local Language Experts and Native Speakers

An intermediate step between planning and creation was to identify and contact any available local experts in the targeted languages, and recruit additional native speakers to serve as annotators and language informants. Annotators were not necessarily linguists or other language experts, but they were native speakers with reading and writing proficiencies who received training as needed from in-house language experts for creating our annotated corpora, and helped to identify and evaluate harvestable online resources.

Intensive recruiting efforts were conducted for native speakers of each language. Our recruiting strategy utilized such resources as online discussion boards and student associations for those language communities, and we were also able to capitalize on the diversity of the student/staff body at the University of Pennsylvania by recruiting through posted signs on campus.

5.2 Identifying Available Language Resources

The first step in creating each language pack was to identify resources that are already available. To this end we implemented a series of "Harvest Festivals"; intensive sessions where our entire team, along with native speaker informants, convened to search the web for available resources. Available resources were immediately documented on a shared and group editable internal wiki page. By bringing together native speakers, linguists, programmers, information managers and projects managers in the same room, we were able to minimize communications latency, brainstorm as a group, and quickly build upon each other's efforts. This approach was generally quite successful, especially for the text corpora and lexicons, and brought us some of our most useful data.

5.3 Investigating Intellectual Property Rights

As soon as Raw digital resources were identified, our local intellectual property rights specialist began investigation into their usability for the REFLEX-LCTL language packs. It was necessary to contact many individual data providers to obtain an agreement, so the process was quite lengthy and in some cases permission was not granted until shortly before the package was scheduled for release to the REFLEX community. Our general practice was to process all likely candidate data pools and remove data as necessary in later stages, thus ensuring that IPR was not a bottleneck in language pack creation. The exception to this was for large data sources, where removal would have significantly affected the quantity of data in the deliverable.

5.4 Creating Basic Text Processing Tools

The next step was to create the language-specific basic data processing tools, such as encoding converter, sentence segmenter and tokenizer.

The goal for this project was to include whatever encoding converters were needed to convert all of

[3]We thank Partha Pratim Talukdar for providing frameworks for creating taggers.

the raw text and lexical resources collected or created into the standard encoding selected for that target language.

Dividing text into individual sentences is a necessary first step for many processes including the human translation that dominated much of our effort. Simple in principle, LCTL sentence segmentation can prove tantalizingly complex. Our goal was to produce a sentence segmenter that accepts text in our standard encoding as input and outputs segmented sentences in the same encoding.

Word segmentation, or tokenization, is also challenging for languages such as Thai. For Thai, our approach was to utilize an existing opensource word segmentation tool, and enhancing it by using a larger word list than the provided one.

We designed the basic format conversion tools, such as the text-to-LTF converter, to be able to just plug in language-specific tokenizers and segmenters.

5.5 Creating Monolingual Text

The monolingual text corpora in the languages packs were primarily created by identifying and harvesting available resources from the Internet, such as news, newsgroups and weblogs in the target language. Once the IPR expert determined that we can use the resources for the REFLEX-LCTL program, we harvested the document files – recent documents as archived documents. The harvested files were then analyzed and the files that actually have usable contents, such as news articles and weblog postings were kept and converted into the LCTL Text format. The formatting process was typically done in the following steps: 1) convert the harvested document or article in html or other web format to a plain text format, stripping html tags, advertisements, links and other non-contents; 2) convert the plain text files into UTF-8, 2) verify the contents with native speakers, and if necessary, further remove non-contents, or divide a file into multiple files; 3) convert the plain text files into the LCTL Text format, applying sentence segmentation and tokenization. Each document file is assigned a unique document ID. Essential information about the document such as the publication date was kept in the final files.

5.6 Creating Parallel Text

Each language pack contains at least 250,000 words of parallel text. Part of this data was found resources harvested from online resources, such as bilingual news web site. The found parallel documents were converted into the LTF format, and aligned at the sentence level, producing segment-mapping tables between the LTF files in the LCTL language and the LTF files in English.

The rest of this data was created by manually translating documents in the LCTL language into English, or documents in English into the LCTL language. A subset of the monolingual text corpus was selected for translation into English.

In addition, about 65,000 words of English source text were selected as the "Common English Subset" for translation into each LCTL language. Having the same set of parallel documents for all languages will facilitate comparison between any or all of the diverse LCTL languages. The Common Subset included : newswire text, weblogs, a phrasebook and an elicitation corpus. The phrasebook contained common phrases used in daily life, such as "I'm here", and "I have to go". The elicitation corpus, provided by Carnegie Mellon University (Alvarez et al., 2006), included expressions, such as "*male_name_1* will write a book for *female_name_1*, where *male_name_1* and *female_name_1* are common names in the LCTL language. The set of elicitation expressions is designed to elicit lexical distinctions which differ across languages.

The manual translation tasks were outsourced to translation agencies or independent translators. Since the translators were instructed to translate text which had already been sentence-segmented, the creation of sentence-level mappings was trivial. However, we found that it was important to create a sentence-numbered template for the translators to use, otherwise we were likely to receive translations where the source text sentence boundaries were not respected.

5.7 Creating Lexicons

Bilingual lexicons are also an important resource that can support a variety of human language technologies, such as machine translation and information extraction. The goal for this resource was a lexicon of at least 10,000 lemmas with glosses and parts of speech for each language. For most of the languages, we were able to identify existing lexicons, either digital or printed, to consult with and extract information for a subset of the lexical entries; however, in all cases we needed to process them substantially before they could be used efficiently. We performed quality checking, normalizing, added parts of speech and glosses, added entries and removed irrelevant entries.

5.8 Creating Annotated Corpora

A subset of the target language text in each language pack received up to three types of annotations: part-of-speech tags, morphological analysis, and named entity tags. Named entity annotations were created for all language packs.

Annotations were created by native speakers using the annotation tools discussed in section 4.

5.9 Creating Morphological Analyzers

To address the requirement to include some kind of tool for morphological analysis in each language pack, we created either a morphological analyzer implementing hand-written rules or a stemmer using an unsupervised statistical approach, such as the approach described in (Hammarstrom, 2006).

5.10 Creating Named Entity Taggers

We created a named entity tagger for each language pack using our common development framework for named entity taggers4.3. The tagger was created using the named entity annotated text we created for the language packs.

5.11 Creating Part-of-Speech Taggers

Similarly, we created a POS tagger for each language pack using our common development framework for POS taggers (See Section 4.3). We coordinated the POS tag sets for the taggers and lexicons.

5.12 Creating Name Transliterators

A transliterator that converts the language's native script into Latin script is a desired resource. For some languages, this is not a straightforward task. For example, not all vowels are explicitly represented in Bengali script, and there can be multiple pronunciations possible for a given Bengali character. Names, especially names foreign to the target language exhibit a wide variety of spelling, and in HLTs, make up a large percentage of the out-of-vocabulary terms. We focused on creating a transliterator to for romanization of names in the LCTL languages. This resource was generally created by the programming team with consultation from native speakers.

5.13 Writing Grammatical Sketches

The grammatical sketches provide an outline of the features of the written language, to provide the language engineers with description of challenges specific to the languages in creating language technologies. These sketches were written mainly by senior linguists in our group, for readers who do not necessarily have training in linguistics. The format of these documents was either html or pdf.

6 Summary of Completed Language Packs

Table 1 summarizes the contents of the 8 Asian language packs .[4] All of the language packs have already been distributed to REFLEX-LCTL participating research sites. The packs continue to be used to develop and test language technologies. For example, the Urdu pack was used to support a task in the 2006 NIST Open MT Evaluation campaign (of Standards and Technology, 2009). Once a language pack has been used for evaluation it will be placed into queue for general release.

[4]The numbers represent the number of tokens.

7 Conclusion

We have developed an efficient approach for creating basic text language resources for diverse languages. Our process integrated the efforts of software programmers, native speakers, language specialists, and translation agencies to identify and built on already available resources, and create new resources as efficiently as possible.

Using our streamlined processes, we were able to complete language packs for eight diverse Asian languages. We hope that the completed resources will provide valuable support for research and technology development for these languages.

We faced various challenges at the beginning of the project which led us to revisions of our methods, and some of these challenges would surely be encountered during a similar effort. We hope that our approach as described here will be of service to future endeavors in HLT development for under-resourced languages.

References

Alison Alvarez, Lori S. Levin, Robert E. Frederking, Simon Fung, and Donna Gates. 2006. The MILE corpus for less commonly taught languages. In *Proceedings of HLT-NAACL 2006.*

ELDA. 2008. BLARK Resource/Modules Matrix. From Evaluations and Language Resources Distribution Agency (ELDA) web site http://www.elda.org/blark/matrice_res_mod.php , accessed on 2/23/2008.

Raymond G. Gordon, Jr., editor. 2005. *Ethnologue: Languages of the World, Fifteenth edition, Online version.* SIL International. http://www.ethnologue.com/.

Harald Hammarstrom, 2006. *Poor Man's Stemming: Unsupervised Recognition of Same-Stem Words.* Springer Berlin / Heidelberg.

Andrew Kachites McCallum. 2002. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu.

National Institute of Standards and Technology. 2009. NIST Open Machine Translation Evaluation. http://www.itl.nist.gov/iad/mig/tests/mt/, accessed on June 7, 2009.

	Urdu	Thai	Bengali	Tamil	Punjabi	Tagalog	Pashto	Uzbek
Mono Text	14,804	39,700	2,640	1,112	13,739	774	5,958	790
Parallel Text (L ⇒ E)	1,300	694	237	308		203	180	206
Parallel Text (Found)	947	1,496	243		230			
Parallel Text (E ⇒ L)	65	65	65	65	65	65	65	65
Lexicon	26	232	482	10	108	18	10	25
Encoding Converter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sentence Segmenter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Word Segmenter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
POS Tagger	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
POS Tagged Text	5	5		59				
Morphological Analyzer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Morph-Tagged Text	11			144				
NE Annotated Text	233	218	138	132	157	136	165	93
Named Entity Tagger	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Name Transliterator	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Descriptive Grammar	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Language Packs for Asian Languages (Data Volume in 1000 Words)

A DTD for LTF Files

```
<!ELEMENT LCTL_TEXT (DOC+)                         >
<!ATTLIST LCTL_TEXT lang CDATA         #IMPLIED
                    source_file CDATA  #IMPLIED
                    source_type CDATA  #IMPLIED
                    author CDATA       #IMPLIED
                    encoding CDATA     #IMPLIED    >

<!ELEMENT DOC (HEADLINE|DATELINE|AUTHORLINE|TEXT)+ >
<!ATTLIST DOC id    ID     #REQUIRED
              lang  CDATA  #IMPLIED
>

<!ELEMENT HEADLINE (SEG+)                          >
<!ELEMENT DATELINE (#PCDATA)                       >
<!ELEMENT AUTHORLINE (#PCDATA)                     >
<!ELEMENT TEXT (P|SEG)+                            >

<!ELEMENT P (SEG+)                                 >

<!ELEMENT SEG (ORIGINAL_TEXT?, TOKEN*)             >
<!ATTLIST SEG id             ID     #REQUIRED
              start_token    IDREF  #IMPLIED
              end_token      IDREF  #IMPLIED
              start_char     CDATA  #IMPLIED
              end_char       CDATA  #IMPLIED
>

<!ELEMENT ORIGINAL_TEXT (#PCDATA)                  >

<!ELEMENT TOKEN (#PCDATA)                          >
<!ATTLIST TOKEN id           ID     #REQUIRED
                attach       (LEFT|RIGHT|BOTH)
                                    #IMPLIED
                pos          CDATA  #IMPLIED
                morph        CDATA  #IMPLIED
                gloss        CDATA  #IMPLIED
                start_char   CDATA  #IMPLIED
                end_char     CDATA  #IMPLIED
>
```