PanLex and LEXTRACT: Translating all Words of all Languages of the World

Timothy Baldwin,♠ Jonathan Pool♡ and Susan M. Colowick♡

♠ CSSE
University of Melbourne
tb@ldwin.net

♡ Utilika Foundation
{pool,smc}@utilika.org

Abstract

PANLEX is a lemmatic translation resource which combines a large number of translation dictionaries and other translingual lexical resources. It currently covers 1353 language varieties and 12M expressions, but aims to cover all languages and up to 350M expressions. This paper describes the resource and current applications of it, as well as LEXTRACT, a new effort to expand the coverage of PANLEX via semi-automatic dictionary scraping.

1 Introduction

Translation dictionaries, multilingual thesauri, and other translingual lexical (more precisely, lemmatic) resources answer queries of the form "Given lemma X in language A, what possible translations of it into language B exist?" However, existing resources answer only a small fraction of the potential queries of this form. For example, one may find attested translations of the Santiago del Estero Quichua word *unta* into German, English, Spanish, Italian, French, Danish, Aymara, and several other Quechua languages, but not into the other (roughly 7 thousand) languages in the world.

Answers to the vast majority of possible lemmatic translation queries must be inferred. If *unta* can be translated into Spanish as *lleno*, and *lleno* can be translated into Hungarian as *tele*, e.g., perhaps Quichua *unta* can be translated into Hungarian as *tele*. But such inference is nontrivial, because lexical ambiguity degenerates the quality of indirect translations as the paths through intermediate languages grow longer.

	Current	Goal
Resources	766	10K
Language varieties	1353	7000
Expressions	12M	350M
Expression–meaning pairs	27M	1000M
Expression–expression pairs	91M	1000M

Table 1: Current and goal PANLEX coverage

Thus, it appears that the quality and range of lemmatic translation would be supported by an easily accessible graph combining a large (or, ideally, complete) set of translations reported by the world's lexical resources. PANLEX (http://panlex.org) is a project developing a publicly accessible graph of attested lemmatic translations among all languages. As of 2010, it provides about 90 million undirected pairwise translations among about 12 million lemmata in over 1,300 language varieties, based on the consultation of over 750 resources, as detailed in Table 1. By 2011 it is expected that the resources consulted will approximately quadruple.

2 The PANLEX Project

PanLex is an attempt to generate as complete as possible a translation graph, made up of expression nodes, meaning nodes, and undirected edges, each of which links an expression node with a meaning node. Each expression is uniquely defined by a character string and a language. An expression e_i is a translation or synonym of an expression e_j iff there is at least one meaning m_k such that edges $v(e_i, m_k)$ and $v(e_j, m_k)$ exist. For example, *frame* in English shares a meaning with *bikar* in Bahasa Malay, and *bikar* shares a meaning with *beaker* in English, but *frame* shares no

Coling 2010: Demonstration Volume, pages 37–40,
Beijing, August 2010

meaning with *beaker*. Whether e_i and e_j are synonyms or translations depends on whether their languages are identical. In Table 1, "expression–meaning pairs" refers to edges $v(e,m)$ and "expression–expression pairs" refers to expressions with at least one meaning in common.

2.1 Current Applications of PANLEX

While lemmatic translation falls short of sentential and discourse translation, it is not without practical applications. It is particularly useful in author–machine collaborative translation, when authors are in a position to lemmatize expressions. The prototype PANIMAGES application (`http://www.panimages.org`), based on PANDICTIONARY, elicits a lemmatic search query from the user and expands the query into dozens of languages for submission to image-search services. Hundreds of thousands of visitors have used it to discover relevant images labeled in languages they do not know, sometimes selecting particular target languages for cultural specificity or to craft less ambiguous queries than their own language would permit (Christensen et al., 2009).

In lemmatic messaging applications developed for user studies, users lemmatized sentences to tell stories or send mail across language boundaries. Even with context-unaware translation of lemmata producing mostly non-optimal translations, users were generally able to reconstruct half or more of the originally intended sentences (Soderland et al., 2009). The PanLex database was also used in a multilingual extension of the image-labeling game initiated by Von Ahn and Dabbish (2004).

User and programmatic interfaces to PanLex are under development. A lemmatic user interface (`http://panlex.org/u`) communicates with the user in a potentially unlimited set of languages, with PanLex dynamically using its own data for the localization. A primitive API makes it possible for developers to provide, or make infrastructural use of, lemmatic translation via PanLex. Prototype lemmatic translation services like TeraDict (`http://panlex.org/demo/treng.html`), InterVorto (`http://panlex.org/demo/trepo.html`), and TümSöz (`http://panlex.org/demo/trtur.html`) exploit the API.

2.2 Extraction and Normalization

The approach taken by PANLEX to populate the translation graph with nodes and edges is a combination of: (a) extraction of translation pairs from as many translingual lexical resources as can be found on the web and elsewhere; and (b) inference of new edges between expressions that exist in PANLEX.

To date, extraction has taken the form of hand writing a series of regular expression-based scripts for each individual dictionary, to generate normalized PANLEX database records. While this is efficient for families of resources which adhere to a well-defined format (e.g. FREEDICT or STARDICT dictionaries), it does not scale to the long tail of one-off dictionaries constructed by lexicographers using ad hoc formats, as detailed in Section 2.2. LEXTRACT is an attempt to semi-automate this process, as detailed in Section 3.

Inference of new translation edges is nontrivial, because lexical ambiguity degenerates the quality of indirect translations as the paths through intermediate languages grow longer. PANDICTIONARY is an attempt to infer a denser translation graph from PANLEX combining translations from many resources based on path redundancy, evidence of ambiguity, and other information (Sammer and Soderland, 2007; Mausam et al., 2009; Mausam et al., 2010).

PANLEX is more than a collection, or docbase, of independent resources. Its value in translation inference depends on its ability to combine facts attested by multiple resources into a single graph, in which lemmata from multiple resources that are substantively identical are recognized as identical. The obstacles to such integration of heterogeneous lexical data are substantial. They include: (1) ad hoc formatting, including format changes between portions of a resource; (2) erratic spacing, punctuation, capitalization, and line wrapping; (3) undocumented and non-standard character encodings; (4) vagueness of the distinction between lemmatic (e.g. *Rana erythraea*) and explanatory translations (e.g. *a kind of tree frog*); and (5) absence of consensus for some languages as to the representation of lemmata, e.g. hyphenation and prefixation in Bantu languages, and inclusion or exclusion of tones in tonal languages.

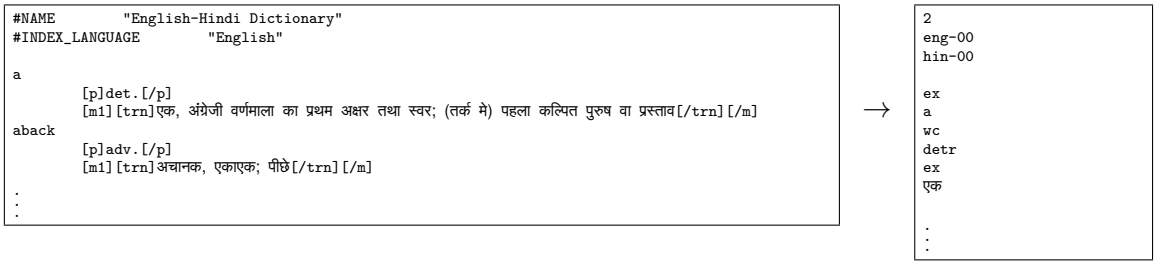

Figure 1: A snippet of an English–Hindi dictionary, in its source form (left) and as normalized PANLEX records (right)

3 LEXTRACT

LEXTRACT is a sub-project of PANLEX, aimed at automating the extraction and normalization of data from arbitrary lexical resources, focusing in the first instance on text-based resources, but ultimately including XML, (X)HTML, PDF and wiki markup-based resources. The approach taken in LEXTRACT is to emulate the manual workflow used by the PANLEX developers to scrape data from dictionary files, namely learning of series of regular expressions to convert the source dictionary into structured database records. In this, we assume that the source dictionary has been transcoded into utf-8 encoding,[1] and further that the first five PANLEX translation records found in the source dictionary have been hand generated as seed instances to bootstrap the extraction process off, as illustrated in Figure 1. Briefly, this provides vital data including: specification of the source and target languages; manual disambiguation of expression–expression vs. expression–meaning structuring; any optional fields such as part of speech; and (implicitly) where the records start from in the source file, and what fields in the original dictionary should not be preserved in the PANLEX database.

The procedure for learning regular expressions can be broken down into 3 steps: (1) record matching; (2) match lattice pruning; and (3) regular expression generalization.

Record matching involves determining the set of codepoint spans in the original dictionary where the component strings (minimally the source and target language expressions, but possibly including domain information, word class information or other metadata) encoded in the five seed records can be found, to use as the basis for learning the formatting idiom employed in the dictionary. For each record, we determine all positions in the source dictionary file where all component strings can be found within a fixed window width of one another. This is returned as a match lattice, representing the possible sub-extents ("spans") in the source dictionary of each record, and the location(s) of each component string within each.

Match lattice pruning takes the match lattice from the record matching step, and prunes it based on a combination of hard and soft constraints. The single hard constraint currently used at present is that the records must occur in the lexicon in sequence; any matches in the lattice which violate this constraint can be pruned. Soft constraints include: each record should span the same number of lines; the fields in each record should occur in the same linear order; and the width of the inter-field string(s) should be consistent. These are expectations on dictionary formatting, but can be violated (e.g. a given dictionary may have some entries on a single line and others spanning two lines). To avoid over-pruning the lattice, we determine the coverage of each such soft constraint in the form of: (a) *type*-level coverage, i.e. the proportion of records for which a given constraint setting (e.g. record size in terms of the number of lines it spans) matches with at least one record span; and (b) *token*-level coverage, i.e. the proportion of individual spans a given constraint setting matches. We apply soft constraints conservatively, selecting the soft constraint setting with full type-level coverage (i.e. it matches all records)

[1] We have experimented with automatic character encoding detection methods, but the consensus to date has been that methods developed for web documents, such as the CHARDET library, are inaccurate when applied to dictionary files.

and maximum token-level coverage (i.e. it prunes the *least* edges in the lattice). Soft constraints are applied iteratively, as indicated in Algorithm 1.

Algorithm 1 Match lattice pruning algorithm

```
Initialize l                          ▷ initialize record matching match lattice
repeat
    change ← False
    for all h_i ∈ H do                ▷ update hard constraint coverage
        (htype_i, htoken_i) ← coverage(h_i, l)
        if htoken_i < 1 then          ▷ if pruneable edges
            l ← apply(h_i, l)         ▷ apply constraint
            change ← True
        end if
    end for
    for all s_i ∈ S do                ▷ update soft constraint coverage
        {(stype_ij, stoken_ij)} ← coverage(c_i, l)
    end for
    if s ← arg max_{s_ij}(∃stype_ij = 1.0 ∧ stoken < 1.0 ∧
    (∀i' ≠ i : |stype_i*| > 1, ∀j' : stoken_ij < 1.0 : stoken_ij >
    stoken_i'j')) then
        l ← apply(s, l)               ▷ apply constraint
        change ← True
    end if
until change = False
```

The final step is regular expression generalization, whereby the disambiguated match lattice is used to identify the multiline span of all records in the source dictionary, and inter-field strings not corresponding to any record field are generalized across records to form a regular expression, which is then applied to the remainder of the dictionary to extract out normalized PANLEX records. As part of this, we build in dictionary-specific heuristics, such as the common practice of including optional fields in parentheses.

The LEXTRACT code is available from `http://lextract.googlecode.com`.

LEXTRACT has been developed over 10 sample dictionaries, and record matching and match lattice pruning has been found to perform with 100% precision and recall over the seed records. We are in the process of carrying out extensive evaluation of the regular expression generalization over full dictionary files.

Future plans for LEXTRACT to get closer to true emulation of the manual extraction process include: dynamic normalization of target language strings (e.g. normalizing capitalization or correcting inconsistent pluralization) using a combination of language-specific tools for high-density target languages such as English, and analysis of existing PANLEX expressions in that language; elicitation of user feedback for extents of the document where extraction has failed, fields where the correct normalization strategy is unclear (e.g. normalization of POS tags not seen in the seed records, as for *det.* →*detr* in Figure 1); and extending LEXTRACT to handle (X)HTML and other file types.

References

Christensen, Janara, Mausam, and Oren Etzioni. 2009. A rose is a roos is a ruusu: Querying translations for web image search. In *Proc. of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 193–196, Suntec, Singapore.

Mausam, Stephen Soderland, Oren Etzioni, Daniel Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proc. of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 262–270, Suntec, Singapore.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, and Jeff Bilmes. 2010. Panlingual lexical translation via probabilistic inference. *Artificial Intelligence*, 174(9–10):619–637.

Sammer, Marcus and Stephen Soderland. 2007. Building a sense-distinguished multilingual lexicon from monolingual corpora and bilingual lexicons. In *Proc. of the Eleventh Machine Translation Summit (MT Summit XI)*, pages 399–406, Copenhagen, Denmark.

Soderland, Stephen, Christopher Lim, Mausam, Bo Qin, Oren Etzioni, and Jonathan Pool. 2009. Lemmatic machine translation. In *Proc. of Machine Translation Summit XII*, page 2009, Ottawa, Canada.

Von Ahn, Luis and Laura Dabbish. 2004. Labeling images with a computer game. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 319–326, Vienna, Austria.