Context-Dependent Multilingual Lexical Lookup for Under-Resourced Languages

**Input:** Text $Q = \text{sequence of lexical items (LIs)} \{ w_1, w_2, \ldots, w_n \}

**Output:** Ranked list of translations sets for each $w_i$

**Motivation: Under-Resourced Languages**
- WSD and translation selection typically require rich lexical knowledge and/or corpus resources
- Gloss texts, subject codes, semantic networks
- Corpora: (un)tagged; mono-/bilingual; aligned/comparable
- Whither under-resourced languages?
  - Lack of rich lexical resources
  - Small corpora

**Comparable Bilingual Corpora**
- **Idea:** leverage from *richer-resourced language pairs*
- Mine translation context data from ‘rich’ language pair
- Comparable corpora is easier to obtain
- E.g. concatenated English & Malay Wikipedia articles
  - 62,993 documents, 67,499 terms

**Multilingual Lexicon**
- **Translation sets**, like multilingual synsets
- Corresponding to coarse-grained concepts
- Member lexical items (LIs) from different languages
  - 24,371 English, 13,226 Chinese, 35,640 Malay, 17,063 French, 14,687 Thai, 5,629 Iban LIs
- Uses translation context data from rich language pair for entire translation set

**Extracting Translation Context Knowledge**
- Document = bilingual article pair as bag-of-words
- Preprocess corpus (segment, stop word removal, lemmatise)
- Run latent semantic indexing (LSI) on corpus
- 100 factors, 45 minutes (MacBook Pro 2.3 GHz, 4 GB RAM)
- One term vector for each LI in both languages
- $V(TS) = \sum (\text{all available term vectors of member LIs})$
- $V(TS_i) = V(\langle \text{bank}\rangle_{\text{eng}}) + V(\langle \text{bank}\rangle_{\text{msa}})$
- $V(TS) = V(\langle \text{bank}\rangle_{\text{eng}}) + V(\langle \text{tebing}\rangle_{\text{msa}})$

**Context-Dependent Lexical Lookup**
- For input text $Q$, $V_Q = \sum V(w_i)$
- If $Q$ language not in training corpus, $V_Q = \sum V(\text{all TS containing } w_i)$
- For each LI $w_i$ in $Q$, $TS_{w_i} = \{ t_1, \ldots, t_n \}$ set of all TS containing $w_i$
- $TS_{w_i}$ sorted by $\text{CSim}(V_i, V_Q) = \frac{\sum V_{t_i} \times V_{t_j}}{|V_i| \times |V_Q|}$

**Example with Iban Input (Top Ranked Translation Sets Shown)**
- Q = ‘Lelaki nya tikah enggau emperaja iya, siko dayang ke ligung’
  - lelaki zho: 男性 (man)
  - tikah zho: 结婚 (marriage)
  - emperaja zho: 情人 (lover)
  - ligung zho: 可爱 (lovely)
- Q = ‘Udah ujian nya ngetu terbubuh, matahi enggau emperaja lalu ayan ba langit’
  - ujian zho: 考试 (exam)
  - ngetu zho: 停止 (stop)
  - matalahi zho: 太阳 (sun)
  - langit zho: 天空 (sky)

**Some Quick Results**
- 80 text sentences: «bank»$_{\text{eng}}$, «plant»$_{\text{eng}}$, «kabinet»$_{\text{msa}}$, «mangga»$_{\text{msa}}$, «sumber»$_{\text{msa}}$, «emperaja»$_{\text{iba}}$
- Strategies
  - wiki-lsi Proposed strategy
  - base-freq Baseline: most frequent translation
  - goog-tr Google Translate
- Metrics: Precision, Mean Reciprocal Rank

**Conclusions**
- Trained on bilingual comparable corpus
- But can be used for *multilingual* inputs
- May not be highly accurate, but *fast, cheap* for under-resourced languages