The TALP N-gram based Machine Translation System

Patrik Lambert

DCU Internal MT Workshop, July 2008
1. N-gram-based Machine Translation
   - N-gram-based Translation Model
   - Full n-gram-based Translation System

2. N-gram-based versus Phrase-based SMT

3. The MARIE decoder

4. N-gram based MT at the NCLT
The translation model

- Translation Model:
  N-gram language model of bilingual units (tuples)

\[
p(S, T) = \prod_{k=1}^{K} p((\tilde{s}, \tilde{t})_k | (\tilde{s}, \tilde{t})_{k-N+1}, \ldots, (\tilde{s}, \tilde{t})_{k-1})
\]

- Tuples are extracted from word alignment
  - A unique, monotonous segmentation of each sentence pair is produced.
  - No word in a tuple is aligned to words outside of it
  - No smaller tuples can be extracted without violating the previous constraints
Tuple extraction

```
NULL | quisieramos | lograr | traducciones perfectas
we   | would like  | to achieve | perfect translations
```

Tuples:
1. - NULL : we
2. - quisieramos : would like
3. - lograr : to achieve
4. - traducciones perfectas : perfect translations

Phrases:
1. - quisieramos : would like
2. - quisieramos : we would like
3. - lograr : to achieve
4. - traducciones : translations
5. - perfectas : perfect
6. - quisieramos lograr : would like to achieve
7. - quisieramos lograr : we would like to achieve
8. - traducciones perfectas : perfect translations
9. - lograr traducciones perfectas : to achieve perfect translations
10. - quisieramos lograr traducciones perfectas :
    would like to achieve perfect translations
11. - quisieramos lograr traducciones perfectas :
    we would like to achieve perfect translations
From the translation model to the translation system

- n-gram-based translation model alone can produce translations, but search is better guided with more models
- \( \hat{T} = \arg \max_T \sum_m \lambda_m h_m(S, T) \) ; Features:
  - Target language model (standard n-gram model)
  - Word bonus model: \( p_{WP}(T) = \exp(\text{number of words in } T) \).
From the translation model to the translation system

- n-gram-based translation model alone can produce translations, but search is better guided with more models
- $\hat{T} = \arg\max_T \sum_m \lambda_m h_m(S, T)$; Features:
  - Target language model (standard n-gram model)
  - Word bonus model: $p_{WP}(T) = \exp(\text{number of words in } T)$.
  - A source-target lexical model, which use IBM1 translation probabilities to compute a lexical weight for each tuple
    $$p_{IBM1}((\tilde{s}, \tilde{t})_k) = \frac{1}{(l+1)^j} \prod_{j=1}^{J} \sum_{i=0}^{I} p(t^i_k | s^j_k)$$
  - A target-source lexical model
- Model weights optimisation: implementation of a tool based on two possible algorithms: Downhill Simplex and SPSA
N-gram-based (NB) vs Phrase-based (PB) SMT

- Similar number of translation units
- Lower decoding time for N-gram based MT (unique vs multiple segmentation of sentence pair)
- PB system not very sensitive to histogram pruning parameter. NB system more sensitive. Interpretation:
  - PB partial hypotheses scored uncontextualized.
  - NB approach: bad sequence of tuples composed of a good initial sequence may cause the pruning of the rest of hypotheses
MARIE decoder

- Ngram-based Statistical Machine Translation decoder: MARIE [Crego et al., 2005]
- Also works for Phrase-based SMT (N=1)
- New version (LIMSI): N-coder: unrestricted input word graph. Many applications (augmenting the input graph):
  - various word segmentations of Chinese
  - various tokenization of Arabic
  - synonyms of input words better modelled in translation model
  - etc.
N-gram based MT at the NCLT

- Easy-to-use-for-NCLT-people N-gram based system and scripts will be available
- If the LIMSI allows it, new “N-coder” will be available to NCLT (at least the binary)