Decoding with Phrase-Based Translation Models

Michael Collins, Columbia University

Phrase-based Translation

An example sentence:

wir müssen auch diese kritik ernst nehmen

A phrase-based lexicon contains phrase entries (f, e) where f is a sequence of one or more foreign words, e is a sequence of one or more English words. Example phrase entries that are relevant to our example:

(wir müssen, we must)

(wir müssen auch, we must also)

(ernst, seriously)

Each phrase (f, e) has a score $g(f, e)$. E.g.,

$$g(f, e) = \log\left(\frac{\text{Count}(f, e)}{\text{Count(e)}}\right)$$

Phrase-based Models: Definitions

- A phrase-based model consists of:
  1. A phrase-based lexicon, consisting of entries (f, e) such as

     (wir müssen, we must)

     Each lexical entry has a score $g(f, e)$, e.g.,

     $$g(\text{wir müssen, we must}) = \log \left(\frac{\text{Count(wir müssen, we must)}}{\text{Count(we must)}} \right)$$

  2. A trigram language model, with parameters $q(w|u, v)$. E.g., $q(\text{also}|\text{we, must})$.
  3. A "distortion parameter" η (typically negative).

Phrase-based Translation: Definitions

An example sentence:

wir müssen auch diese kritik ernst nehmen

- For a particular input (source-language) sentence $x_1 \ldots x_n$, a phrase is a tuple (s, t, e), signifying that the subsequence $x_s \ldots x_t$ in the source language sentence can be translated as the target-language string e, using an entry from the phrase-based lexicon. E.g., $(1, 2, \text{we must})$
- $\mathcal{P}$ is the set of all phrases for a sentence.
- For any phrase p, $s(p)$, $t(p)$ and $e(p)$ are its three components. $g(p)$ is the score for a phrase.

Definitions

- A derivation y is a finite sequence of phrases, $p_1, p_2, \ldots p_L$, where each p_j for $j \in \{1 \ldots L\}$ is a member of $\mathcal{P}$.
- The length L can be any positive integer value.
- For any derivation y we use $e(y)$ to refer to the underlying translation defined by y. E.g.,

  $y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

  and

  $$e(y) = \text{we must also take this criticism seriously}$$

Valid Derivations

- For an input sentence $x = x_1 \ldots x_n$, we use $\mathcal{Y}(x)$ to refer to the set of valid derivations for x.
- $\mathcal{Y}(x)$ is the set of all finite length sequences of phrases $p_1 p_2 \ldots p_L$ such that:
  - Each p_k for $k \in \{1 \ldots L\}$ is a member of the set of phrases $\mathcal{P}$ for $x_1 \ldots x_n$.
  - Each word in x is translated exactly once.
  - For all $k \in \{1 \ldots (L-1)\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where $d \geq 0$ is a parameter of the model. In addition, we must have $|1 - s(p_1)| \leq d$

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

Examples

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 2, we must), (7, 7, take), (3, 3, also), (4, 5, this criticism), (6, 6, seriously)

Scoring Derivations

The optimal translation under the model for a source-language sentence x will be

$$\arg\max_{y \in \mathcal{Y}(x)} f(y)$$

In phrase-based systems, the score for any derivation y is calculated as follows:

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

where the parameter η is the distortion penalty (typically negative). (We define $t(p_0) = 0$).

$h(e(y))$ is the trigram language model score. $g(p_k)$ is the phrase-based score for p_k.

An Example

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

Decoding Algorithm: Definitions

- A state is a tuple

$$(e_1, e_2, b, r, \alpha)$$

where e_1, e_2 are English words, b is a bit-string of length n, r is an integer specifying the end-point of the last phrase in the state, and α is the score for the state.

- The initial state is

$$q_0 = (*, *, 0^n, 0, 0)$$

where 0^n is bit-string of length n, with n zeroes.

States, and the Search Space

wir müssen auch diese kritik ernst nehmen

$$(*, *, 0000000, 0, 0)$$

Transitions

- We have $ph(q)$ for any state q, which returns set of phrases that are allowed to follow state $q = (e_1, e_2, b, r, \alpha)$.
- For a phrase p to be a member of $ph(q)$, it must satisfy the following conditions:
  - p must not overlap with the bit-string b. I.e., we need $b_i = 0$ for $i \in \{s(p) \ldots t(p)\}$.
  - The distortion limit must not be violated. More formally, we must have $|r + 1 - s(p)| \leq d$ where d is the distortion limit.

An Example of the Transition Function

wir müssen auch diese kritik ernst nehmen

$(\textsf{must, also}, 1110000, 3, -2.5)$

An Example of the Transition Function

wir müssen auch diese kritik ernst nehmen

$(\text{must, also}, 1110000, 3, -2.5)$

In addition, we define $next(q, p)$ to be the state formed by combining state q with phrase p.

The $next$ function

Formally, if $q = (e_1, e_2, b, r, \alpha)$, and $p = (s, t, \epsilon_1 \ldots \epsilon_M)$, then next$(q, p)$ is the state $q' = (e_1', e_2', b', r', \alpha')$ defined as follows:

- First, for convenience, define $\epsilon_{-1} = e_1$, and $\epsilon_0 = e_2$.
- Define $e_1' = \epsilon_{M-1}$, $e_2' = \epsilon_M$.
- Define $b_i' = 1$ for $i \in \{s \ldots t\}$. Define $b_i' = b_i$ for $i \notin \{s \ldots t\}$
- Define $r' = t$
- Define

$$\alpha' = \alpha + g(p) + \sum_{i=1}^{M} \log q(\epsilon_i | \epsilon_{i-2}, \epsilon_{i-1}) + \eta \times |r + 1 - s|$$

The Equality Function

- The function

$$\mathsf{eq}(q, q')$$

  returns true or false.

- Assuming $q = (e_1, e_2, b, r, \alpha)$, and $q' = (e'_1, e'_2, b', r', \alpha')$, $eq(q, q')$ is true if and only if $e_1 = e'_1$, $e_2 = e'_2$, $b = b'$ and $r = r'$.

The Decoding Algorithm

- Inputs: sentence $x_1 \dots x_n$. Phrase-based model $(\mathcal{L}, h, d, \eta)$. The phrase-based model defines the functions $ph(q)$ and next(q, p).
- Initialization: set $Q_0 = \{q_0\}$, $Q_i = \emptyset$ for $i = 1 \dots n$.
- For $i = 0 \dots n - 1$
  - For each state $q \in$ beam(Q_i), for each phrase $p \in ph(q)$:
    (1) $q' =$ next(q, p)
    (2) Add(Q_i, q', q, p) where $i =$ len(q')
- Return: highest scoring state in Q_n. Backpointers can be used to find the underlying sequence of phrases (and the translation).

An Example

wir müssen auch diese kritik ernst nehmen

$$(*, *, 0000000, 0, 0)$$

Definition of Add(Q, q', q, p)

- If there is some $q'' \in Q$ such that $eq(q'', q') =$ True:
  - If $\alpha(q') > \alpha(q'')$
    - $Q = \{q'\} \cup Q \setminus \{q''\}$
    - set $bp(q') = (q, p)$
  - Else return
- Else
  - $Q = Q \cup \{q'\}$
  - set $bp(q') = (q, p)$

Definition of beam(Q)

Define

$$\alpha^* = \arg\max_{q \in Q} \alpha(q)$$

i.e., α^* is the highest score for any state in Q.

Define $\beta \geq 0$ to be the *beam-width* parameter
Then

$$\text{beam}(Q) = \{q \in Q : \alpha(q) \geq \alpha^* - \beta\}$$