Node Alignment Algorithm

AVENUE Group Rule Learner Re-Write
Greg Hanneman and Jonathan Clark

July 20, 2010

1 Definitions of Coverage

We have several notions of the coverage of a node. In the following definitions, *vectors* are implemented as bit sets while *spans* are implemented as a pair of integers. Nodes are arranged in a source-side parse tree covering source sentece $w_1 ... w_n$ and a target-side parse tree covering target sentence $w'_1 ... w'_m$. The notation $X \rightsquigarrow y$ indicates that node X dominates terminal word y in a parse tree. The source and target sentence are word aligned; an alignment between source word w_i and target word w'_j is shown as $w_i \leftrightarrow w'_j$.

- **Direct coverage vector of a source node $b(S)$.** For a node in the source-side parse tree, the contiguous range of source word indexes covered by the yield of the node. The direct coverage vector $b(S)$ is defined as:

$$b(S) = \{i : S \rightsquigarrow w_i\} \tag{1}$$

- **Direct coverage vector of a target node $b(T)$.** Similar to the above, but for target-side nodes and target word indexes.

$$b(T) = \left\{j : T \rightsquigarrow w'_j\right\} \tag{2}$$

- **Projected coverage vector of a source node $\beta(S)$.** For a node in the source-side parse tree, the exact target word indexes covered by the node via projection through the word alignments. For a source terminal node, the projected coverage vector comes directly from the word alignments; for a non-terminal node, it is the union of the projected coverage vectors of its children.

$$\beta(S) = \begin{cases} \left\{j : S \rightsquigarrow w_i, w_i \leftrightarrow w'_j\right\} & \text{if } S \text{ terminal} \\ \bigcup_{k \in \text{children}(S)} \beta(k) & \text{otherwise} \end{cases} \tag{3}$$

- **Projected coverage span of a source node $\beta_{min}(S)$, $\beta_{max}(S)$.** Similar to the above, but only the overall minimum and maximum of the target word indexes are kept.

$$\beta_{min}(S) = \min\{j : j \in \beta(S)\} \tag{4}$$

$$\beta_{max}(S) = \max\{j : j \in \beta(S)\} \tag{5}$$

- **Projected complement vector of a source node $\bar{\beta}(S)$.** For a node in the source-side parse tree, the exact target word indexes *not* covered by the node and its descendants. For the source root node, the projected complement vector is a vector of all 0s in the length of the target sentences. For other source nodes S, the projected complement vector $\bar{\beta}(S)$ is defined as:

$$\bar{\beta}(S) = \bar{\beta}\left(\text{parent}(S)\right) \cup \bigcup_{k \in \text{siblings}(S)} \beta(k) \tag{6}$$

The notation is apt to get quite dense, but here is its motivation: b denotes the coverage bit set of a node, while the equivalent Greek letter β denotes a projection of that bit set. Complement bit sets are denoted with a bar: $\bar{b}$ or $\bar{\beta}$. It is important to remember, however, that the bar syntax does not indicate a logical `NOT` operation: β and $\bar{\beta}$ are not simply bitwise negations of each other.

2 T2T Alignment

2.1 Procedure

1. Read in the source tree, target tree, and Viterbi alignments between them. When the source tree is read in, compute the direct coverage vector of each source node. When the target tree is read in, compute the direct coverage vector of each target node.

2. In a bottom-up procedure, compute the projected coverage vector and projected coverage span of each source and target node. For a terminal node, the projected coverage vector is the set of target word indexes the source terminal is aligned to. For a non-terminal node, the projected coverage vector is the union of projected coverage vectors of the node's children. (See Equation 3.) For any node, the projected coverage span is the minimum and maximum indexes of bits set in the projected coverage vector.

3. In a top-down procedure, compute the projected complement vector of each source node. For the root node, the projected complement vector is initialized to all 0s. For other nodes, the projected complement vector is computed as defined in Equation 6.

4. Discover aligned nodes:

   (a) For a node s, if the intersection of its projected coverage span and its projected complement vector is 0, then a node alignment can be made in T2S mode. (In T2S alignmnent, the string-side span s is aligned to is $[\beta_{min}(s), \beta_{max}(s)]$.)

   $$\beta(s) \cap \bar{\beta}(s) = \mathbf{0} \Longrightarrow s \text{ consistent} \tag{7}$$

   (b) To find out what the exact node alignment is for T2T mode, find an exact match between s's projected coverage vector and a target node t's direct coverage vector. This check must be made in both directions to avoid unaligned boundary words. Because projected coverage vectors do not include internal unaligned words, also construct vectors u and u' of unaligned source and target words and use them to fill in the gaps within projected coverage vectors. The notation $v[a, b]$ means the subsequence of vector v from position a to position b (inclusive).

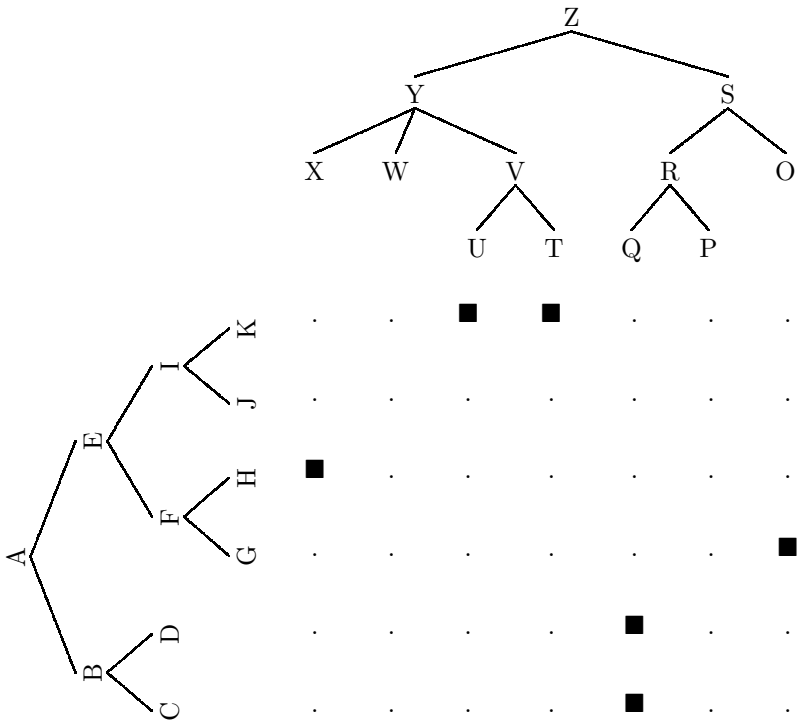
   $$\left.\begin{array}{l} \beta(s) \cup u'[\beta_{min}(s), \beta_{max}(s)] = b(t) \\ \beta(t) \cup u[\beta_{min}(t), \beta_{max}(t)] = b(s) \end{array}\right\} \Longrightarrow s \text{ aligned to } t \tag{8}$$

   (c) To find "grown" node alignments for s in T2T mode, find target nodes t where the union of s's projected coverage vector and a vector u' of all target-side unaligned words exactly equals the union of t's direct coverage vector and a vector of all target-side unaligned words.

   $$\beta(s) \cup u' = b(t) \cup u' \Longrightarrow s \text{ aligned to } t \tag{9}$$

2.2 Example

We word through the following example, where the tree headed at A is the source and the tree headed at Z is the target. Black squares indicate word alignments links between terminal nodes.



1. The trees are read in and the direct coverage vectors are computed. For example, node E spans words 3 through 6 in the source tree, so its direct coverage vector is $b(E) =$ <001111>.

Node	$b(s)$
A	<111111>
B	<110000>
C	<100000>
D	<010000>
E	<001111>
F	<001100>
G	<001000>
H	<000100>
I	<000011>
J	<000010>
K	<000001>

Node	$b(t)$
Z	<1111111>
Y	<1111000>
X	<1000000>
W	<0100000>
V	<0011000>
U	<0010000>
T	<0001000>
S	<0000111>
R	<0000110>
Q	<0000100>
P	<0000010>
O	<0000001>

2. Projected coverage vectors and spans are computed bottom-up for the source tree. For example, source terminal node G is aligned to the word 7 in the target sentence, so its projected coverage vector is $\beta(G) =$ <0000001> and its projected coverage span is $(\beta_{min}(G), \beta_{max}(G)) = (7, 7)$. Likewise, source terminal node H is aligned to word 1 in the target tree, so $\beta(H) =$ <1000000> and $(\beta_{min}(H), \beta_{max}(H)) = (1, 1)$. At the non-terminal node F, the projected coverage vector is the union of its children's vectors, $\beta(F) =$ <1000001>. Thus, the projected coverage span for F is $(\beta_{min}(F), \beta_{max}(F)) = (1, 7)$.

Node	$\beta(s)$	β_{min}, β_{max}
A	<1011101>	1, 7
B	<0000100>	5, 5
C	<0000100>	5, 5
D	<0000100>	5, 5
E	<1011001>	1, 7
F	<1000001>	1, 7
G	<0000001>	7, 7
H	<1000000>	1, 1
I	<0011000>	3, 4
J	<0000000>	$\emptyset$, $\emptyset$
K	<0011000>	3, 4

3. Projected complement vectors are computed top-down for the source tree. For example, source node F has parent E and sibling I, so the projected complement vector at F is $\bar{\beta}(F) = \bar{\beta}(E) \cup \beta(I)$. At this point in the computation $\bar{\beta}(E)$ has already been set to <0000100>, and $\beta(I)$ in the previous step was found to be <0011000>. Therefore, $\bar{\beta}(F) =$ <0000100> $\cup$ <0011000>, or <0011100>.

Node	$\bar{\beta}(s)$
A	<0000000>
B	<1011001>
C	<1011101>
D	<1011101>
E	<0000100>
F	<0011100>
G	<1011100>
H	<0011101>
I	<1000101>
J	<1011101>
K	<1000101>

4. Node alignments are discovered in three stages:

   (a) Nodes are checked for consistency by comparing their projected coverage spans and projected complement vectors. For example, source node F has projected coverage span $(\beta_{min}(F), \beta_{max}(F)) = (1, 7)$, which is cast to a vector $\beta_{min...max}(F) =$ <1111111>. F also has projected complement vector $\bar{\beta}(F) =$ <0011100>. The intersection of these two vectors is <0011100>, which is non-zero, so F is not consistently aligned.

Node	$\beta_{min..max}(s) \cap \bar{\beta}(s)$	**Consistent?**
A	<0000000>	Yes
B	<0000000>	Yes
C	<0000100>	No
D	<0000100>	No
E	<0000100>	No
F	<0011100>	No
G	<0000000>	Yes
H	<0000000>	Yes
I	<0000000>	Yes
J	< $\emptyset$ >	No
K	<0000000>	Yes

   (b) The consistent source nodes are aligned to target nodes by exact match of (1) a source node's projected coverage vector union all target-side unaligned words within that span with a target node's direct coverage vector, and (2) the target node's projected coverage vector union all source-side unaligned words within that span with the source node's direct coverage vector. For example, source node A has projected coverage vector $\beta(A) =$ <1011101>. The unaligned target words

vector between positions 1 and 7 is $u'[1,7] =$ <0100010>. On the target side, node Z has direct coverage vector $b(Z) =$ <1111111>. Thus, $\beta(A) \cup u'[1,7] =$ <1111111> $= b(Z)$. In the reverse direction, target node Z has projected coverage vector $\beta(Z) =$ <111101>, and the unaligned source words vector between 1 and 6 is $u[1,6] =$ <000010>. The union of these two vectors exactly equals the direct coverage vector of $b(A)$, or <111111>. Therefore, nodes A and Z are aligned.

Node	**Alignments**
A	Z
B	Q
G	O
H	X
K	V

(c) "Grown" alignments for consistent nodes are calculated. A vector u' of all unaligned words in the target tree is calculated and included via union on both sides of the exact match test between a projected coverage vector and a direct coverage vector. For example, $u' =$ <0100010>. Now, at source node B, $\beta(B) \cup u' =$ <0100110>. On the target side there are two matches: $b(R) \cup u' =$ <0100110> and $b(Q) \cup u' =$ <0100110>. Thus, node B now aligns to both node Q and node R.

Node	**Alignments**
A	Z
B	Q, R
G	O
H	X
I	V
K	V