Corpus Navigator Architecture

Jonathan Clark

Thursday, November 29, 2007

1 Building the Search Space

Algorithm 1: Building the search space.
Input: TODO... It will be a long, rather boring list.
Output: The best feature structure that should be elicited next.

```
SearchLanguage($A, B$)
(1)   $q \leftarrow$ empty priority queue
(2)   let $h$ represent a hypothesis
(3)   foreach $s \in elicitedSentences$
(4)       $h \leftarrow s$
(5)       $realization(h) \leftarrow uneliciatedRealizations(h) \cup NOTelicitedRealizations(h)$
(6)       $q \leftarrow q \cup h$
(7)       foreach $x \in (clauses(s) \cup participants(s) \cup modifiers(s) \cup conjunctions(s))$
(8)           $h \leftarrow s - x$ // remove clause/participant
(9)           $q \leftarrow q \cup (h, score(h))$
(10)          foreach feature $f \in featureSetFor(x)$
(11)              foreach featureValue $v \in f \neq f(s)$
(12)                  $h \leftarrow s$
(13)                  $f(h) \leftarrow v$
(14)                  $realization(h) \leftarrow uneliciatedRealizations(h)$
(15)                  $q \leftarrow q \cup (h, score(h))$
(16)      // NOTE: the main clause is not compatible with itself
(17)      foreach $x \in ((allCompatibleClauses \cup allCompatibleParticipants \cup allCompatibleModifiers \cup
          allCompatibleConjunctions) \notin (clauses(s) \cup participants(s)))$
(18)          foreach $y \in seedSubFeatureStructures(x)$
(19)              $hypothesish \leftarrow s + y$
(20)              $realization(h) \leftarrow uneliciatedRealizations(h)$
(21)              $q \leftarrow q \cup (h, score(h))$
(22)  return $top(q)$
```

- NOTE: We can either build all hypotheses first and then rank later so that we can show a progress indicator or
- NOTE: We may have duplicate hypotheses in the search space, but this shouldn't matter; Alternatively, we can build the search space in terms of all combinations instead of in terms of minimal pairs.
- NOTE: We will build hypotheses for sentences that have already been elicited, but we will specify that they must have a different realization.

- NOTE: Since modifiers can occur infinitely many times per clause/participant and conjunctions can have infinitely many items, we might want to limit these to 2 and 3 each, respectively.

Upper bound on number of hypotheses for each state is:
O(numElicitedSentences · (maxNumCompatibleClauses! · numClauses · (maxNumClauseSeeds + numClauseFeatures · maxClauseFeatureValues · maxNumCompatibleParticipants! · numParticipants · (maxNumParticipantSeeds + numNounFeatures · maxNumNounFeatureValues))))

–OR–

The total number of combinations possible, whichever is less. Therefore, our maximum number of hypotheses should be under 100 million, if previous estimates are correct.

1.1 Using Seeds to Make the Search Space Tractable

Consider the case of a participant with 3 features, each having 3 possible values. If we were to allow all combinations to be hypotheses, we would have 27 new hypotheses just in this contrived example. However, if the user can recommend a reasonable starting place (seed), then we reduce this to a single (or perhaps 2 or 3) hypotheses, preventing the combinatoric explosion.

2 Scoring the Hypotheses

Input here is the set of unscored hypotheses from above h. We then score them all and take the argmax. This is a theoretical formulation, since we might actually prune some of the hypotheses with small scores if the search space grows too large.

The Score function is defined as the following mixture model: If we had a gold standard, we could theoretically run Expectation Maximization on the mixing weights of our scoring metrics.

(Should we make this log-linear?)

Score(k=currentKnowledge, f=featureStructure, m=metaContext, l=lexicalElements) =
$\lambda_0 \cdot booleanExploreIf(k, v, m)$
$\cdot($
$\lambda_1 \cdot \sum_{v \in f}$
$\cdot($

// General FEG's
$\lambda_a \cdot \sum_{i=2}^{numInteractions+1} weightForInterationLevel(i) \cdot \sum_{x \in affectedFEGs(k,v,m,i)}$

$\sum_{y \in assertedArcs(k,v,m,i,x)} likelihoodOfInteraction(y) \cdot redundancyFunction[y](numAlreadyExplored(y))/(i \cdot numAlreadyExplored(y))$

// Specific FEG's
$+ \lambda_b \cdot \sum_{i=2}^{numInteractions+1} weightForInteractionLevel(i) \cdot \sum_{w \in metaElements} \sum_{x \in affectedFEGs(k,v,m,i)}$

$\sum_{y \in assertedArcs(k,v,m,i,x)} likelihoodOfInteraction(y)$

// General Single Features
$+ \lambda_3 \cdot likelihoodOfExpression(v)$

// Specific Single Features
$+ \lambda_4 \cdot likelihoodOfExpression(v) \cdot metaContextWeight(v, m)$

$+ \lambda_5 \cdot numFeatureValuesRemainingToBeExpanded(feature(v))$
$+ \lambda_6 \cdot percentFeatureValuesRemainingToBeExpanded(feature(v))$
)

$+ \lambda_7 \cdot vocabularyCoverage(f, l)$
$+ \lambda_8 \cdot morphologyGoodness(k, f, m, l)$
$+ \lambda_9 \cdot goodnessOfStayingInThisSeedOrSwitching$
$+ \lambda_{10} \cdot haveRealizedSentence(f)$
)

Can we formulate this as a Bayesian model? A n-gram model similar to MT? Add value for task and frequency of occurrence in target domain/open domain.

2.1 Counting Crossings, Reorderings, and Discontinuities

targetDiscontinuity(X) $= 0$ if $(S_i \rightarrow T_j), \forall T_j, x \leq j \leq y, x = \text{argmin}_b(S_a \rightarrow T_b), y = \text{argmax}_b(S_a \rightarrow T_b), \{S_i, S_a\} \subseteq$ children(X)
else 1

3 Doing Inference on the Collected Evidence

*** This section is still under construction. ***

We can label the expression of a feature value v as:

1. *fully expressed* if its minimal pairs are all have different target language translations
2. If any other value is expressed within this feature $f \ni v$, then we say v is *jointly expressed* with the other values that can be clustered with it.
3. *unexpressed* if all feature values are identical.

t_i are translated sentences
f_i are feature structures
x, y are feature values
$x \vdash f$ indicates the substitution of another feature value y for x into a feature structure f where y is in the same feature as x. For example, we might substitute *singular* for *plural*.

$P_x(\Delta(t_1, t_2) \mid x \in f_1 \cap f_2 \equiv x \vdash f_1)$
$=$

$P_{x,y}(\Delta(t_1, t_2) \mid x, y \in f_1 \cap f_2 \equiv y \vdash f_1)$
$= P_{x,y}(\Delta(t_1, t_2) \mid y \in f1, f2 \cap x \in f_1 \cap f_1 \equiv x \vdash f_2)$
$=$

NOTE: We are NOT finding dependence in the statistical sense, but instead just finding implications. Also, we report statistics rather than estimate probabilities.

Algorithm 2: Inferring expression of single features and the relationships between them.
Input: TBA... It will be a longish list.
Output: A list of feature expressions and relationships along with collected statistics.

```
InterRelations($A, B$)
(1)
    let $\theta$ be a user-defined threshold value $e \leftarrow$ empty set of expression values foreach $a \in$ features
(2)     foreach $x \in$ values($a$)
(3)         $p \leftarrow P_x(\Delta(t_1, t_2) \mid x \in f_1 \cap f_2 \equiv x \vdash f_1)$ if $p \geq \theta$
(4)             // TODO: Write clustering code
(5)     $t \leftarrow$ empty key-value table $(\{x, y\}, r)$ where $\{x, y\}$ is an unordered set of feature values and $r$ is a set of relationships
(6)     foreach $a \in$ features
(7)         foreach $b \in$ features
(8)             foreach $x \in$ values($a$)
(9)                 foreach $y \in$ values($b$)
(10)                    // add dependency for whichever direction creates fewer rules
```

4 Jargon

1. Feature Value Dominance -
2. Realization - The process of writing source language sentences given a feature structure.
3. Participants - Roles defined in a feature structure. e.g. Actor, Undergoer, Causer, Causee
4. Clause Types - e.g. main-clause, relative-clause, etc.
5. Inference - The process of determining which features are actually expressed in the target language.
6. Feature Interaction Level - The number of features that we will simultaneously examine for interactions between each other. I plan to start with 2.
7. Feature Expression Graph (FEG) - A graph containing one node for each feature combination being analyzed (during inference). When analyzing zero feature interactions, An arc is drawn between nodes to represent one or more pieces of elicited evidence for the existence of a difference between the two feature value combinations.
8. General FEG - A FEG that does not include participants or clauses as part of its combinations.
9. Specific FEG - A FEG that does include participants and/or clauses as part of its combinations.
10. Elicitation Search - For us, the process of determining which sentences to feed to the inference engine.
11. Hypothesis - A candidate feature structure for elicitation whose goodness will be scored and ranked. A hypothesis might or might not be associated with a realized source language string.
12. Seed Sentences - An “opening book” of feature structures with specified realizations that will always be explored during the search process. These seeds serve as baseline from which minimal pairs will be created.
13. Partial Feature Structure - An extracted portion (much like a constituent) of a feature structure. For example, we might extract the partial feature structure associated with just a relative clause just an actor.
14. Partial Seeds - Partial feature structures used for pruning the combinatoric explosion associated with adding a new clause/participant to a sentence.
15. Minimal Pair - During inference, During search, a minimal pair is made up of one sentence that has has already been elicited and a hypothesis.

16. Scoring Metric - A function used to determine the goodness of one particular aspect of a hypothesis.

17. Evaluation Metric - A function used to determine the performance of the system as a whole.

18. Mixing Weights / Lambdas - Weights used to combine the scoring metrics into a single score value.

19. Rule Implication - The “then” part of a rule in the production system.

20. Implicational Universal - A Greenbergian Typological Universal, most likely computed on the WALS feature set rather than on our MILES features.

21. Feature Relationships - Language facts that is the output of the inference stage of our system.