Supplementary Material: ILP Encoding of the Tree Generation Step in Sentence Enhancement

Jackie Chi Kit Cheung
University of Toronto
10 King's College Rd., Room 3302
Toronto, ON, Canada M5S 3G4
jcheung@cs.toronto.edu

Gerald Penn
University of Toronto
10 King's College Rd., Room 3302
Toronto, ON, Canada M5S 3G4
gpenn@cs.toronto.edu

ILP Encoding Details

This document pertains to the paper *Unsupervised Sentence Enhancement for Automatic Summarization*. Below, we describe the implementation details of the novel objective function and syntactic constraints in the integer linear programming formulation of sentence generation.

A Informativeness score

The word informativeness function $I(a)$ depends on the frequency of a word a and the syntactic depth at which a is found in the source text in terms of the level of embedding by the number of clause boundaries crossed from the root of the tree:

$$I(a) = \frac{\mathit{depth}(a)}{\mathit{max\_depth}} \mathit{freq}(a) \times \log \frac{F_{\mathit{ALL}}}{F_a}, \qquad (1)$$

where *depth*(a) is the syntactic depth of a in the parse tree of the sentence by the number of clause boundaries crossed on the path to the root of the tree, *max_depth* is the maximum depth of the sentence, *freq*(a) is the frequency of word a in the document cluster, F_{ALL} is the total count of content words in the entire corpus, and F_a is the total count of word a in the corpus. The first factor captures the intuition that words that are more deeply embedded tend to be more important, at least in news text, where there are many uninformative reporting verbs near the top of a parse tree such as *said* or *announced*. The other factors are an instantiation of TF-IDF as a method of determining term importance.

B Objective function

Our objective function is designed to maximize word and syntactic relation importance while avoiding redundancy in the tree that is extracted from the expanded sentence graph. Let X be the set of variables in the linear program, and let each variable in X take the form $x_{h,r,a}$, a binary variable that represents whether an edge in the sentence graph from a head node with lemma h to an argument with lemma a in relation r is selected. For a lexicon Σ, our objective function is:

$$\max \sum_{w \in \Sigma} \max_{x_{h,r,a} \in X s.t. a=w} (x_{h,r,w} \cdot P(r|h) \cdot I(w)) \qquad (2)$$

Since this is an integer linear program, the inner $\max$ must be factored out of the objective function by the introduction of auxiliary variables and constraints. First, we introduce an auxiliary variable $y_{h,r,a}$ for each original variable $x_{h,r,a}$. Call the set of these auxiliary variables Y. We rewrite the objective function in terms of these auxiliary variables, removing the inner max function:

$$\max \sum_{y_{h,r,a} \in Y} y_{h,r,a} \cdot P(r|h) \cdot I(a). \qquad (3)$$

We then add constraints in order to relate the auxiliary variables to their corresponding original variable, and to ensure that each lemma is only scored once. For the former, we constrain the auxiliary variables to be at most the value of the original:

$$y_{h,r,a} \leq x_{h,r,a}. \qquad (4)$$

Then, we add a constraint for each lemma w in the lexicon Σ, such that at most one auxiliary variable may be "on" for each lemma:

$$\forall w \in \Sigma, \sum_{y_{h,r,a} \in Y s.t. a=w} y_{h,r,w} \leq 1. \qquad (5)$$

The modified objective is equivalent to the original if the program is solved optimally, as the auxiliary variables will be set such that only the highest scoring $y_{h,r,a}$ variable for each lemma a contributes a positive value to the objective function.

C Syntactic constraints

Nominal and adjectival predicate In Stanford's collapsed dependency representation, nominal and adjectival predicates are indicated by a *nsubj* relation from the predicate head to the argument, and a *cop* relation to the copular, usually some form of the verb *to be*. We add a constraint to ensure these pairs are selected together, and another to ensure that the construction is found at the top level of a finite clause.

Transitive verbs In order to ensure transitive verbs take both of their expected arguments, we need to implement the constraint for each relevant node that the number of dependents with the relation *nsubj* is greater than 0 if and only if the number of *dobj* children is greater than 0.

For a particular transitive verb node n in the expanded sentence graph, let the sets of variables in X that represent the *nsubj* children be denoted as $X_{n,nsubj}$. Then, we introduce a variable $h_{n,nsubj}$ that has value 1 if and only if at least one variable in $X_{n,nsubj}$ is 1:

$$\forall x \in X_{n,nsubj}, h_{n,nsubj} \geq x \tag{6}$$

$$h_{n,nsubj} \leq \sum_{x \in X_{n,nsubj}} x. \tag{7}$$

We likewise introduce constraints for $X_{n,dobj}$ and $h_{n,dobj}$. Then, we simply enforce that:

$$h_{n,nsubj} = h_{n,dobj}. \tag{8}$$

D Semantic constraints

We followed F&S in disallowing noun phrases that are in a hyponym/hyperonym or holonym/meronym relation from being coordinated, as indicated by WordNet. We also disallowed noun phrases whose heads are dissimilar, according to the distributional semantic model described in Section 3.2.1. Here, "dissimilar" means the cosine similarity falls below the observed average of conjunct similarity scores in the corpus, which was 0.3317. Rather than embed these constraints into the ILP as F&S, we precomputed the results, and simply added a constraint to the ILP to disallow conjunction between each pair of nodes that may not be conjoined.