Title of Work:

Human-like Natural Language Generation Using Monte Carlo Tree Search

Conference:

INLG (International Language Generation Conference) 2016 Workshop on Computational Creativity and Natural Language Generation

Rationale to ensure venue quality:

The INLG is an international, biennial conference of the Special Interest Group on Natural Language Generation (SIGGEN), and is sponsored by the ACL, ARRIA, and the WebNLG Organization. It has ties to the several research institutions throughout Europe, including The University of Edinburgh and Edinburgh Napier University. This many connections to multiple well-known organizations and universities seems to suggest a high level of quality for the venue.

Problem Statement:

Current Natural Language Generation techniques often fail to account for the situational nature of speech. Both the syntactic structure and content of a spoken sentence can be affected or made ambiguous by context, making proper NLG or searching difficult. Making use of a Monte Carlo Tree Search, inputting grammar rules as search operators, can allow for a more effective generation of sentences that appropriately match a context.

Paper Synopsis:

The article begins by describing some of the issues inherent to speech that make Natural Language Generation difficult – people unconsciously produce utterances in daily life based on different situations. These utterances are created with words appropriate to react to a given stimulus or context. In order for NLG Systems to become truly effective, they must be able to mimic this ability to dynamically generate utterances in response to stimuli. However, this ability is intrinsically very difficult for computer systems, as combining the right words and vocabulary while maintaining syntactic correctness is a very complex task. Kumagai et al propose utilizing a Monte Carlo tree search, which is a "stochastic search algorithm for decision processes," to build a search tree of possible syntactic trees that we can use to generate a sentence.

A Monte Carlo tree search (MCTS) combines random simulation and best-first search, making use of an upper confidence bounds value to determine the next "move to make". The System builds a search space of potential steps for a syntactic tree (such as splitting a subject into a Noun Phrase and Verb Phrase), and uses context-free grammar rules from the *Brown Corpus* as a search operator to iteratively determine which grammar rule to apply to grow the tree by simulating multiple potential derivations of that tree.

To evaluate their results, Kumagai et al score the generated sentences on its syntactic structure and on an n-gram language model. To evaluate the syntactic structure, a classifier was built using logistic regression to identify if a sentence is natural or not. 4,661 sentences of three to seven words were extracted, parsed, and used to create a set of context free grammars that contained 7,220 grammar rules and 5,867 terminal symbols. 46,610 syntactically incorrect sentences were generated as negative examples to show syntactically incorrect sentences (the disparity in sentences models that syntactically incorrect sentences are much more likely than syntactically correct sentences). FREQuent Tree miner was used to extract and analyze syntactic subtrees and return a classification result. To evaluate the n-gram language model (essentially, the validity of the sequence of words), two more metrics are used. The first is a calculation of the perplexity of trigrams with Kneser-Ney smoothing and the second is a term called "acceptability," which measures the acceptability of a sentence for an English native speaker. These scores are aggregated to quantify how valid a sequence of words is in a sentence.

The final step is ranking the "final decision" as a pass/fail metric. In order to emphasize the importance of valid syntax, any sentence that failed the syntactic evaluation automatically failed and given a score of 0. If it passed the syntactic evaluation, but not the n-gram language model evaluation, it was given low passing score of 0.5. Finally, passing both evaluations netted a full score of 1.0.

Future Work:

The main area of future work I would to see explored is extended these tests and classifier generation across multiple information domains, building a larger suite of classifiers and exploring how different domains affect language structure. Additionally, I would like to see the n-gram language model evaluated on different criterion than just trigrams and perplexity. An experiment that accounts for different n-grams or applying an F-score on search retrieval results could offer further insight into the validity of sentences generated by the application