Ant Colony Algorithm Applied to Automatic speech Recognition Graph Decoding

Benjamin Lecouteux, Didier Schwab

Ant Colony Algorithm Applied to Automatic speech Recognition Graph Decoding

Benjamin Lecouteux and Didier Schwab

LIG Université Grenoble-Alpes http://www.liglab.fr/
GETALP Team http://getalp.imag.fr

Introduction

We propose an original approach that allows the decoding of ASR Graphs by using a constructive algorithm based on ant colonies. In classical approaches, when a graph is decoded with higher order LM ; the algorithm must expand the graph in order to develop each new observed n-gram. This extension process increases the computation time and memory consumption. We propose to use an ant colony algorithm in order to explore ASR graphs with a new language model, without the necessity of expanding it.

Related work

Classical approaches :

- **Synchronous graph algorithms** : generate a virtual copy of the graph for all hypotheses and that terminate after a fixed time
- **Reentrant graph algorithms** : a virtual copy of the graph is explored for every linguistic context.
- **Stack asynchronous algorithms** : deep exploration and prioritize promising hypotheses.
- The rescoring of a 2-gram graph with 3-gram language model increases drastically the number of nodes.

In order to remove the language model expansion phase, we propose to use an ant colony algorithm to explore the graph.

Ant colony algorithm :

- Comes from the observation of ant social behavior. Ants have the ability to collectively find the shortest path between their nest and a source of energy.
- Interactions are often very simple and allow the colony to solve complex problems (swarm intelligence).
- Ant colony algorithms are a good alternative for the resolution of problems modeled by graphs. They allow a fast and efficient exploration that exhibits performance close to other search methods.
- Such algorithms have already been applied for natural language processing tasks successfully including Word Sense Disambiguation.

ASR system, used corpora and baseline results

ASR system based on the KALDI toolkit

- Acoustic model trained on LIUM corpora : 118 hours of annotated data. Δ, ΔΔ coefficient parameters and LDA, MLLT are applied. SAT+fMLLR adaptation is used.
- LM trained on data provided for the IWSLT 2010 campaign as well as the training corpora provided by LIUM.

LM order	# n-grams	Perplexity	dev WER	test WER
2g	+ 35M	234	22.01%	21.67%
3g	+ 238M	159	18.1%	17.7%
4g	+ 524M	150	17.4%	16.7%

TABLE: *Details on the different language models. (lexicon : 41K words)*

Used Corpora and baseline results

- Experiments performed on the TED corpus 2010 that corresponds to a set of conference talks recorded in English.
- We purposely used an initial bi-gram language model in order to not introduce additional information at the linguistic level in the ASR.
- Then the graphs were rescored without pruning with 3 and 4-gram language models.

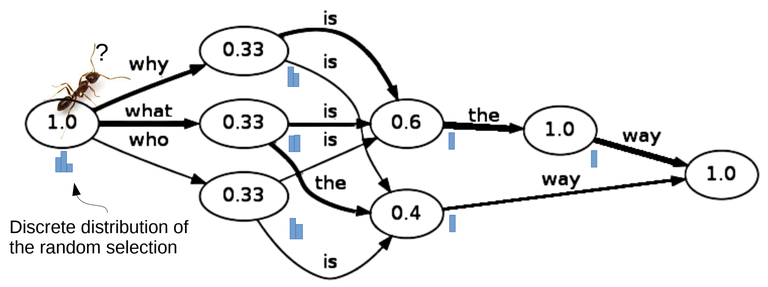
TED	# Sentences	#	duration	3g WER	4g WER
dev	507	18226	4h12	18.1 %	17.4 %
test	1155	28430	7h30	17.7 %	16.7 %

TABLE: *Details on TED corpora used for the experiments.*

Proposed approaches

ASR : We use an initial bi-gram language model, so that we could use the ant colony algorithm to expand the graph to higher n-gram orders.

Algorithm : Each link of the graph is associated with a pheromone variable initialized at the beginning of each epoch. Our ants are launched from the first node of the graph and in order to leave a node, ants will have to choose a link. This choice is performed with a weighted random selection based on the quantity of pheromone on each arc. With the arrival of an ant at the endpoint, if a solution is found that is better than the previous best solution found, then pheromone is added on the path.



Experiments

Set	Vit. no prune	Vit. 1xRT	Ant 1 run	10 run	100 runs
Dev 3g	18.1	20.2	18.5	18.4	18.3
Dev 4g	17.4	19.1	17.8	17.7	17.7
Tst 3g	17.7	19.8	18.0	17.9	17.8
Tst 4g	16.7	18.5	17.2	17.0	17.0
Mem.3g	1 GB	200MB	100MB	=	=
Mem.4g	3 GB	500MB	100MB	=	=
Time3g	32h	4h40	0h30	4h40	45h
Time4g	45h	4h40	0h30	4h40	45h
xRT	7&10	1	0.1	1	10

TABLE: *We report the decoding time (dev+test), the memory footprint (average for one graph) and the WER compared to Viterbi beam-search baseline. We compare decoding time vs performance.*

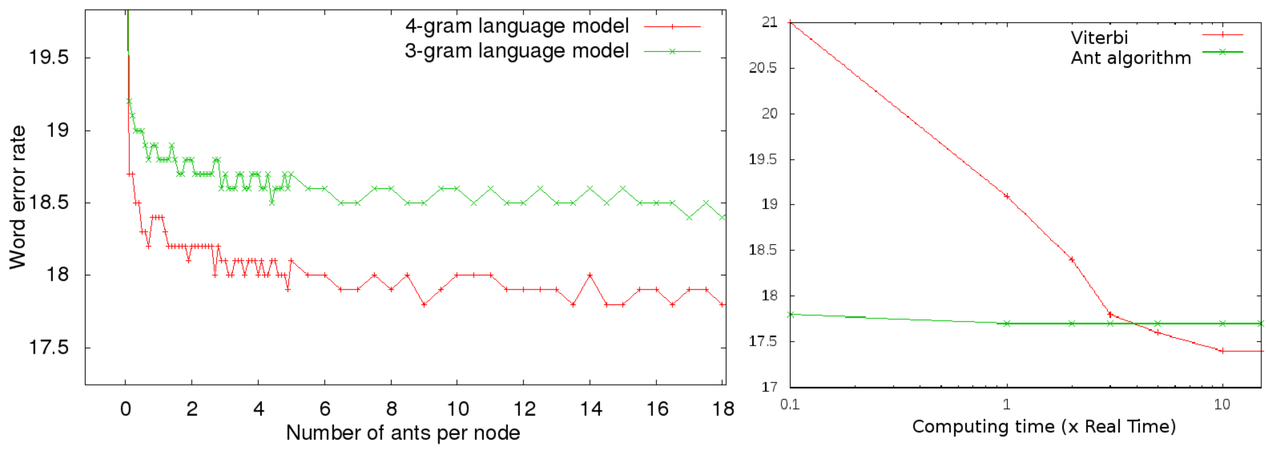


FIGURE: *Evolution of rescoring according the number of ant by node (on the dev set) during 1 run and comparison between Viterbi beam-search and ant colonies algorithm : Computing time against performance, on the dev set with a 4-gram language model. With time constraints the ant algorithm is more efficient than a conventional extension..*

Discussion and Future Works

- Parallelization is trivial : ants can evolve independently of each other.
- Computational time is controlled.
- There are no physical extensions : memory usage is constant.
- The evaluation is performed when ant arrives at the final node.
- We wish to explore and analyze this ant colony paradigm more in depth.
- Use of ants with different behaviors.
- We would like to apply ant colony algorithms throughout the ASR process : from the phoneme lattice to the language model expansion.

INTERSPEECH 2015
September 6 - 10
Dresden, Germany

Contacts: **Benjamin.Lecouteux@imag.fr and Didier.Schwab@imag.fr**