Automated Slogan Production Using a Genetic Algorithm

Polona Tomašič*
XLAB d.o.o.
Pot za Brdom 100
1000 Ljubljana, Slovenia
(polona.tomasic@gmail.com)

Gregor Papa*
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
(gregor.papa@ijs.si)

Martin Žnidaršič*
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
(martin.znidarsic@ijs.si)

*Affiliated also to the Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia

The Student Workshop on Bioinspired Optimization Methods and their Applications (BIOMA 2014)

13 September 2014, Ljubljana, Slovenia

Investing in your future
OPERATION PART FINANCED BY THE EUROPEAN UNION
European Social Fund

Introduction

- Slogan generation – field of Computational Creativity.
- The state of the art: The BRAINSUP framework for creative sentence generation (Özbal, 2013):
  - User provides keywords, domain, emotions, ...
  - Beam search through the search space of possible slogans.
- Our method aims at a completely autonomous approach:
  - User provides only a short textual description of the target entity.
  - Based on a genetic algorithm.
  - Follows the BRAINSUP framework in the initial population generation phase.
  - Uses a collection of heuristic slogan functions in the evaluation phase.

Resources

- The database of existing slogans.

Guinness is good for you.	Guinness
The best a man can get.	Gillette
Say it with flowers.	FTD
Capitalist tool.	Forbes
Hand-built by robots.	Fiat Strada
Put a tiger in your tank.	Esso
The pause that refreshes.	Coca Cola
It's the real thing.	Coca Cola
Probably the best lager in the world.	Carlsberg
I'd walk a mile for a Camel.	Camel
Have it your way.	Burger King
It's good to talk.	British Telecom
The world's favourite airline.	British Airways
The ultimate driving machine.	BMW
Reach out and touch someone.	AT&T
Don´t leave home without it.	American Express

Resources

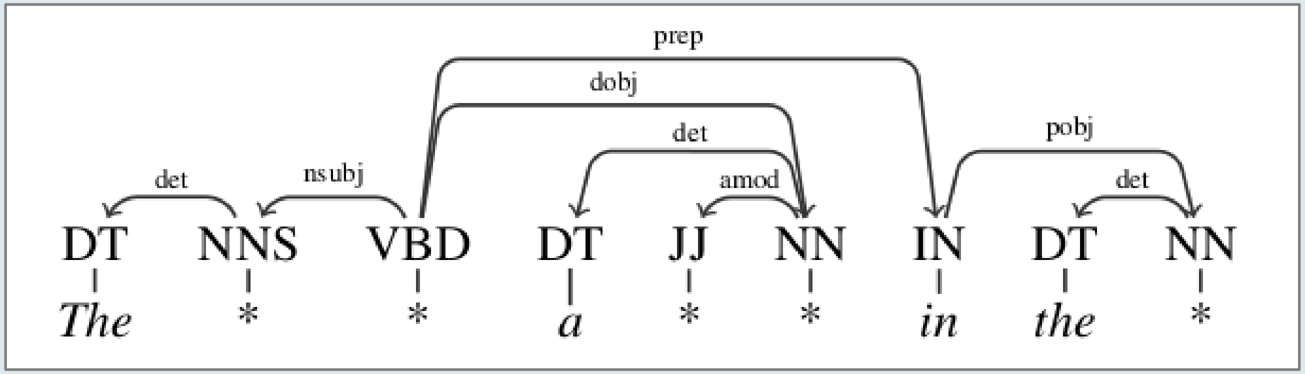
- The database of frequent grammatical relations between words in sentences, along with the part-of-speech tags.

```
Jane is walking her new dog in the park.

(ROOT
  (S
    (NP (NNP Jane))
    (VP (VBZ is)
      (VP (VBG walking)
        (NP (PRP$ her) (JJ new) (NN dog))
        (PP (IN in)
          (NP (DT the) (NN park)))))
    (. .)))

nsubj(walking-3, Jane-1)
aux(walking-3, is-2)
root(ROOT-0, walking-3)
poss(dog-6, her-4)
amod(dog-6, new-5)
dobj(walking-3, dog-6)
det(park-9, the-8)
prep_in(walking-3, park-9)
```

Stanford Dependencies Parser's output for the sentence
"Jane is walking her new dog in the park"

Resources

- The database of slogan skeletons – existing slogans without the content words.



Skeleton

Slogan Generation

INPUT: a textual description of a company or a product and the algorithm parameters

OUTPUT: a set of generated slogans

Algorithm 1 SloganGenerator

Input: A textual description of a company or a product T, Size of the initial population S_{IP}, Maximal number of iterations Max_iter, Crossover probability $p_{\mathrm{crossover}}$, Mutation probability p_{mutation}, Set of evaluation weights W.

Output: A set of generated slogans S.

$Keywords, Entity \Leftarrow$ GetKeywordsAndEntity(T)
$P \Leftarrow$ CreateInitialPopulation($S_{\mathrm{IP}}, Keywords, Entity$)
Evaluate(P)
while $Max\_iter > 0$ **do**
    ChooseParentsForReproduction(P)
    Crossover($P, p_{\mathrm{crossover}}$)
    Mutation(P, p_{mutation})
    DeleteSimilarSlogans(P)
    while Size(P) $< S_{\mathrm{IP}}$ **do**
        AddARandomSeed(P)
    end while
    Evaluate(P)
    $Max\_iter \Leftarrow Max\_iter - 1$
end while
$S \Leftarrow P$

Keywords and Entity Extraction

- **Keywords**: the most frequent nonnegative words in the input text.
- **Entity**: the most frequent entity in the input text.

keywords = ['win', 'celebrate', 'enjoy', 'follow', 'available', 'raspberry', 'snowy', 'cherry', 'famous', 'wonderful', 'familiar', 'sugar', 'sparkle', 'passion', 'beloved', 'fountain', 'bubble', 'enjoyment', 'drink', 'fluid', 'diet', 'candy', 'tour', 'beverage', 'contribution', 'dream', 'vision', ...]
entity = Coke

Keywords and entity extracted from the Coca-Cola Wikipedia page.

Initial Population

Based on the BRAINSUP framework, with some modifications and additions.

Algorithm 2 CreateInitialPopulation

Input: Size of the initial population S_{IP}, a set of target keywords K, and the target entity E.

Output: A set of initial slogans S.

$S \Leftarrow \emptyset$
while $S_{\mathrm{IP}} > 0$ **do**
  $SloganSkeleton \Leftarrow$ SelectRandomSloganSkeleton()
  while not AllEmptySlotsFilled($SloganSkeleton$) **do**
    $EmptySlot \Leftarrow$ SelectEmptySlotInSkeleton($SloganSkeleton$)
    $Fillers \Leftarrow$ FindPossibleFillerWords($EmptySlot$)
    $FillerWord \Leftarrow$ SelectRandomFillerWord($Fillers$)
    FillEmptySlot($SloganSkeleton, FillerWord$)
  end while
  AddFilledSkeleton($S, SloganSkeleton$)
  $S_{\mathrm{IP}} \Leftarrow S_{\mathrm{IP}} - 1$
end while

Evaluation

- An aggregate evaluation function, composed of 9 sub-functions:
  - Bigram function
  - Length function
  - Diversity function
  - Entity function
  - Keywords function
  - Word frequency function
  - Polarity function
  - Subjectivity function
  - Semantic relatedness function
- Changing the weights for tuning the outputs.

Production of a New Generation

- 10% elitism.
- 90% roulette wheel.
- Crossover with a probability $p_{crossover.}$
- Mutation with a probability $p_{mutation.}$
- Deletion of similar slogans.
- Random seeds if necessary.

Production of a New Generation

Crossover

- *Small* crossover:

Parents:
Just [RB] do [VB] it [PRP].
Drink [VB] more [JJR] milk [NN].

Children:
Just drink it.
Do more milk.

- *Big* crossover:

Parents:
We [PRP] bring [VBP] good [JJ] things [NNS] to [DT] life [NN].
Fly [VB] the [DT] friendly [JJ] skies [NNS].

Children:
We bring friendly skies.
Fly the good things to life.

Mutation

- *Small* mutations:
  - replacement of a word with its synonym, antonym, meronym, hyponym, hypernym, or holonym.
- *Big* mutations:
  - deletion of a word,
  - addition of an adjective or an adverb,
  - replacement of a word with another random word with the same part-of-speech tag.

Deletion of Similar Slogans

- Removing duplicate slogans.
- Similar slogans: removing the one with the lower evaluation score.
- Preventing quick convergence of slogans.

Experiments

Input:

a textual description of Coca-Cola, obtained from the Wikipedia.

Algorithm parameters:

- Weights: [bigram: 0.22, length: 0.03, diversity: 0.15, entity: 0.08, keywords: 0.12, frequent words: 0.1, polarity: 0.15, subjectivity: 0.05, semantic relatedness: 0.1]
- $p_{crossover} = 0{,}8$
  ($p_{crossover\_big} = 0.4$, $p_{crossover\_small} = 0.2$, $p_{crossover\_both} = 0.4$)
- $p_{mutation} = 0.7$
  ($p_{mutation\_small} = 0.8$, $p_{mutation\_big} = 0.2$)
- Number of iterations of genetic algorithm: 150
- Number of runs of the algorithm for the same input parameters: 20
- The population size: 25, 50 and 75

Results

Table 1. Comparison of average slogans' scores for sizes of initial population: 25, 50 and 75. (F = Final Slogans, IP = Initial Population)

	Minimum	*Maximum*	*Average*	*Median*	*Standard Deviation*
IP (25)	0.000	0.720	0.335	0.442	0.271
IP (50)	0.000	0.721	0.318	0.377	0.270
IP (75)	0.000	0.736	0.311	0.412	0.270
F (25)	0.542	0.874	0.736	0.754	0.089
F (50)	0.524	0.901	0.768	0.775	0.082
F (75)	0.497	0.920	0.778	0.791	0.086

Results

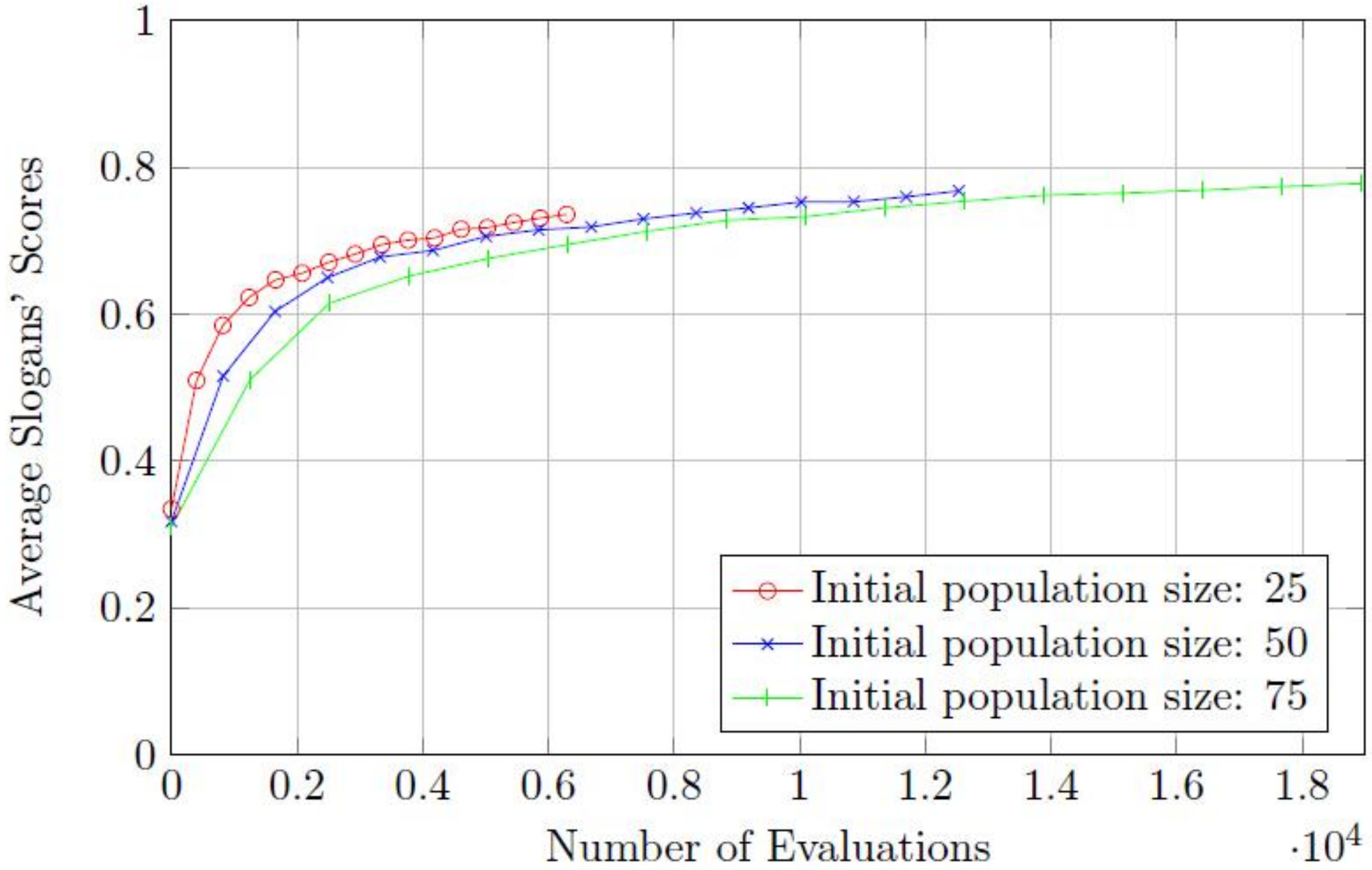


Figure 3. Average Slogans' Scores in Relation to the Number of Evaluations.

Example of final slogans

Size of population: 25

1. Love to take the Coke size (0.906)
2. Rampage what we can take more (0.876)
3. Love the man binds the planetary Coke (0.870)
4. Devour what we will take later (0.859)
5. You can put the original Coke (0.850)
6. Lease to take some original nose candy (0.848)
7. Contract to feast one's eyes the na keep (0.843)
8. It ca taste some Coke in August (0.841)
9. Hoy despite every available larger be farther (0.834)
10. You can love the simple Coke (0.828)

Conclusion and Further Work

- Genetic algorithm ensures the increase of slogan scores with every new generation.
- Current method can be useful for brainstorming.
- Plenty of room for improvement.

Further work:

- refinement of the evaluation functions,
- correction of grammatical errors,
- machine learning for computing the weights of the evaluation functions,
- adaptive calculation of control parameters for genetic algorithm,...

Thank you