

“Talent, Skill and Support.”

A Method for Automatic Creation of Slogans

Khalid Alnajjar
khalid.alnajjar@helsinki.fi

Hadaytullah Hadaytullah
hadaytullah.hadaytullah@helsinki.fi

Hannu Toivonen
hannu.toivonen@helsinki.fi

Department of Computer Science and HIIT, University of Helsinki, Finland

Abstract

Slogans are an effective way to convey a marketing message. In this paper, we present a method for automatically creating slogans, aimed to facilitate a human slogan designer in her creative process. By taking a target concept (e.g. a computer) and an adjectival property (e.g. creative) as input, the proposed method produces a list of diverse expressions optimizing multiple objectives such as semantic relatedness, language correctness, and usage of rhetorical devices. A key component in the process is a novel method for generating nominal metaphors based on a metaphor interpretation model. Using the generated metaphors, the method builds semantic spaces related to the objectives. It extracts skeletons from existing slogans, and finally fills them in, traversing the semantic spaces, using the genetic algorithm to reach interesting solutions (e.g. “Talent, Skill and Support.”). We evaluate both the metaphor generation method and the overall slogan creation method by running two crowdsourced questionnaires.

Introduction

Rhetorical devices are ubiquitous, they are used in daily communications, news, poems, and advertising. This paper focuses on slogans; more specifically, it tackles the task of creating slogans computationally. Slogans are memorable short phrases that express an idea about a product, and are commonly used in advertising campaigns.

In advertising, it is essential to construct expressions wisely. A research conducted by Reece, Van den Bergh, and Li (1994) suggests that recalling a slogan relies mainly on the slogan itself, not on the advertising budget, years in use or themes. Constructing such novel and interesting expressions is a time-consuming task for humans and a challenging one for computers. The method proposed in this paper aims at facilitating the process of constructing such creative expressions by suggesting inspirational slogan candidates tailored to user’s desire. As a result, creative professionals (e.g. writers, advertisers, etc.) can collaborate with computers to produce creative results more efficiently.

Rhetorical devices in slogans have different effects on consumers (Burgers et al. 2015). In this paper, inspired by the work of Miller and Toman (2014), we focus on the two most common rhetorical devices found in slogans:

(1) metaphors and (2) prosody. Miller and Toman have analysed 239 slogans and discovered that 92% of them contained at least one rhetorical device. Tom and Eves (1999)’s research has found that slogans containing rhetorical devices are more persuasive and have higher recall than those that do not.

Our method accepts a target concept and an adjectival property as an input. In advertising, the target concept and adjectival property would be the product type (e.g. a car) and the desired property that the slogan should express (e.g. elegant or luxurious). The method commences by generating apt metaphors for attributing the input property to the target concept. Thereafter, it creates expressions, slogans in our case, adapted to the input and the generated metaphors. A genetic algorithm is employed in the method to search for interesting slogans in the space of possible solutions.

Metaphors consist of two concepts, a tenor and a vehicle following Richards (1936) terminologies, where some properties get highlighted or attributed to the tenor from the vehicle. For instance, in the nominal metaphor “Time is money”, *valuable*, a property of the vehicle *money*, is highlighted in the tenor, *time*. In this paper, the process of metaphor generation targets producing suitable vehicle candidates for expressing the intended adjectival property while considering the input concept.

We also examine the effect of using a corpus-based metaphor interpretation model in generating metaphors. Moreover, we argue that slogans with balanced features (e.g. relatedness to the input and metaphoricity) are comparatively more creative than those with a single dominating feature.

The remainder of this paper is structured as follows. We first briefly review the related work on generating metaphors and rhetorical expressions. Thereafter, we give an overview of resources used by the method. We then describe the method for (1) generating metaphors and (2) generating slogans. Finally, we present the evaluations of our methods and discuss the results.

Related Work

In this section, we review the related work on two computational topics: (1) generation of metaphors and (2) generation of slogans and other creative expressions.

Generation of Metaphors

For the scope of this paper, we review two approaches for generating metaphors.

The first approach, by Xiao and Blat (2013), is focused on generating metaphors for pictorial advertisements. Their approach utilises multiple knowledge bases, e.g. word associations and common-sense knowledge¹, to find concepts with high imageability. The found concepts are then evaluated against four metrics, which are affect polarity, salience, secondary attributes and similarity with tenor. Concepts with high rank on these measures were considered apt vehicles to be used metaphorically.

Galvan et al. (2016) generated metaphors by using a web service, *Thesaurus Rex* (Veale and Li 2013), that provides categorizations of concepts and adjectival properties associated with them. Their approach starts by retrieving top 40% categories of the input tenor. It then selects an adjectival property, at random, that is associated with the tenor. Thereafter, it sends another query to the web service to obtain categories associated with the previously selected property. A category matching the retrieved categories of the tenor is selected. Finally, it creates a metaphor by finding a concept falling in the selected category which is also strongly associated with the selected property.

In contrast to the reviewed metaphor generation methods, our method employs a metaphor interpretation model to identify apt metaphors.

Generation of Creative Expressions

Strapparava, Valitutti, and Stock (2007) proposed a creative function for producing advertising messages automatically. Their approach is based on the “optimal innovation hypothesis” (Giora 2003). The hypothesis states that the optimal innovation is reached when novelty co-exists with familiarity, which encourages the recipient to compare what is known with what is new resulting in a pleasant surprise effect. The approach proposed by the authors utilizes semantic and emotional relatedness along with assonance measures to find interesting candidates of words to substitute some existing words in human-made familiar expressions.

Özbal, Pighin, and Strapparava (2013) have introduced a framework, *BrainSup*, for creative sentence generation. The framework generates sentences such as slogans by producing expressions with semantically related content to the target domain, emotion and colour, and some phonetic properties. The generated expressions must contain keywords that are input by the user. Using syntactical tree-banks of existing sentences as sentence skeletons and syntactical relations between words as constraints for possible candidate fillers, Özbal et al. have employed beam search to greedily fill in the skeletons with candidates meeting the desired criteria.

Using *BrainSup* as a base, Tomašić, Znidaršić, and Papa (2014) have proposed an approach for generating slogans using genetic algorithms instead of beam search. Moreover, their evaluation criteria were different from *BrainSup*'s evaluation. Tomašić et al.'s work demonstrated how it is possible to automatically generate slogans without any user de-

fining target words by extracting keywords from the textual description of the target concept.

Regarding figurative language generation, *Figure8*, by Harmon (2015), generates metaphoric sentences. Five criteria were considered in the generation process, namely: clarity, novelty, aptness, unpredictability, and prosody. The system selects a tenor and searches for a suitable vehicle to express it. Thereafter, it composes sentences to express the metaphor by filling templates of metaphorical and simile expressions.

Our proposed method for generating expressions differs from existing methods as follows. It focuses on generating slogans for a product while expressing a single adjectival property. We want the property to be expressed indirectly and metaphorically. Furthermore, our method creates slogans whilst considering one skeleton at a time. Producing metaphorical expressions is addressed in *Figure8*, which in contrast is concentrated on similes.

Resources

This section covers the linguistic resources used in the proposed methods.

Corpus, ζ We use a 2 billion word web-based text corpus, *ukWaC*², as the main corpus. All corpus-based models in our approach are built using this corpus. We chose a web-based corpus to cover wide range of topics and different writing styles.

Language model, ξ We build a probabilistic bigram language model ξ using bigram frequencies provided with *ukWaC*. The language model is built to estimate the probability of a created slogan to be generated by ξ . A slogan with high probability is more likely to be grammatically correct as it appeared frequently in the corpus ζ . Employing bigrams, in contrast to trigrams or higher n -grams, gives the method a greater degree of freedom in its generations. Higher n -grams would improve the grammar of the generated expressions but would tie them to expressions in the original corpus.

Semantic model, ω We follow the approach described in *Meta4meaning* (Xiao et al. 2016) in building the semantic model ω . The goal of constructing this model is to find words that are semantically related to another word. We start by obtaining co-occurrence counts of words in ζ , constrained by sentence boundaries, within a window of ± 4 . We limit the vocabulary of the model to the most frequent 50,000 words, excluding closed class words. We then convert co-occurrence counts to relatedness measure by employing the log-likelihood measure defined by (Evert 2008) while capping all negative values to zero. Finally, we normalize relatedness scores using L1-norm (McGregor et al. 2015).

¹ConceptNet: <http://www.conceptnet.io>

²<http://wacky.sslmit.unibo.it>

Expression skeletons, δ A slogan skeleton is a parse tree of a phrase where all content words are replaced with a placeholder “***”, i.e. stop words are kept. Nevertheless, all part-of-speech tags (e.g. VBZ) and grammatical relations (e.g. nsubj) between words are retained. The goal of using a database of skeletons is to reuse syntactical structures of effective slogans. The practice of reusing existing slogans can be observed in some well-known slogans, e.g. *Volkswagen’s* “Think Small.” and *Apple’s* “Think Different.”

We utilise *Spacy*³ as a natural language processing tool to parse 40 well-known slogans⁴. Prior to constructing the skeletons, we preprocess the obtained slogans to increase the parsing accuracy. The first preprocessing step is converting capitalized words into lower case, except the first word and any recognized named entities. This step reduces miss-classifying verbs as nouns, yet they could occur as many slogans are not complete sentences. Slogans tend to be informal; therefore, we convert words with the suffix *VERB-in’* into *VERB-ing*, in the second step. As a result of the preprocessing phase, *KFC’s* slogan “Finger Lickin’ Good.” becomes “Finger licking good.”

Subsequently, we convert slogans into skeletons. Figure 1 provides an example of a skeleton generated from *Visa’s* slogan “Life flows better with Visa.”

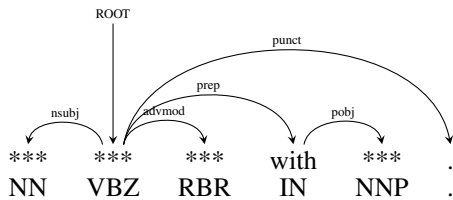


Figure 1: An example of a skeleton constructed from *Visa’s* slogan: “Life flows better with Visa.”

Once all slogans are transformed into skeletons, we only keep skeletons that have at least 40% of their tokens as placeholders and have a minimum of two placeholders. These conditions ensure that the method has some freedom in filling in the skeleton. As a result, slogans such as *Reebok’s* “I am what I am.” and *Coca-Cola’s* “Enjoy.” are removed. In total, the database contained 26 unique skeletons.

Grammatical relations, γ Similarly to approaches by Özal, Pighin, and Strapparava (2013) and Tomašić, Znidaršić, and Papa (2014), we build a repository of grammatical relations. We parse the entire corpus ζ using *Spacy* and store all grammatical relations observed along with their frequencies. A grammatical relation contains a word (called dependent), its head word (called governor), the parts-of-speech of both words, and the type of relation. We retain grammatical relations with frequencies ≥ 50 to remove rare cases. The process yields 3,178,649 grammatical relations.

³<http://www.spacy.io>

⁴Obtained from: <http://www.advergize.com>

Nouns and Their Adjectival Properties, κ We employ two resources for retrieving nouns associated with the input property. The first resource, $\kappa_{General}$, is *Thesaurus Rex* (Veale and Li 2013). *Thesaurus Rex* is used for retrieving general nouns (e.g. coffee, flower, ... etc). On the other hand, the resource provided by Alnajjar et al. (2017), κ_{Human} , is employed to obtain nouns of human categories (e.g. actor, lawyer, politician, ... etc). These resources will be used in generating metaphors, the former for general metaphors and the later for personifications.

Method

In this section, we describe the proposed method. The input to the method is a target concept, T , and an adjectival property, P . An example of such input is $T = \text{“computer”}$ and $P = \text{“creative”}$.

The proposed method is broken into two processes, (1) metaphor generation and (2) slogan generation.

Generation of Metaphors

We define the metaphor generation task as follows. Given a tenor T and an adjectival property P , the generator produces vehicle candidates, $V = \{v_0, v_1, \dots, v_i\}$. A vehicle highlights the adjectival property P in T when perceived metaphorically. An example vehicle candidate for expressing that a *computer* is *creative* is *poet*.

For the input property P , the method begins by retrieving nouns associated with P using κ . We retrieve two types of nouns from the resource κ , general nouns from $\kappa_{General}$ and nouns of human categories from κ_{Human} . We use the top 10% of each type to only pick candidates strongly related to P .

The above procedure gives nouns related to the given property P , but it does not ensure that their metaphorical interpretation in the context of tenor T is P . To select nouns that are likely to have the intended interpretation, we employ a corpus-based metaphor interpretation model, *Meta4meaning* (Xiao et al. 2016).

Meta4meaning accepts two nouns as input, a tenor and a vehicle, and produces a list of possible interpretations for the metaphor. To our knowledge, the proposed method here is the first for generating metaphors based on their interpretations.

Using *Meta4meaning*, the method interprets the potential metaphorical nouns retrieved by calculating the combined metaphor rank metric, c.f. Xiao et al. (2016). Only nouns with the property P among the top 50 interpretations are used. Additionally, as metaphors are asymmetrical, the approach removes vehicle candidates that have the interpretation rank of “ T is [a] v ” greater than to the interpretation of the reversed metaphor, i.e. “ v is [a] T ”.

For example, nouns in κ that are strongly associated with $P = \text{“creative”}$ are:

$\kappa_{General}(\text{creative}) = \{\text{painting, music, \dots, presentation}\}$

$\kappa_{Human}(\text{creative}) = \{\text{artist, genius, poet, \dots, dancer}\}$

By interpreting these candidates using *Meta4meaning* and pruning out candidates not meeting the predefined conditions, we obtain the following candidates where the score

is the interpretation rank:

$V_{General}(computer, creative) = \{\text{art: 4, drama: 4, director: 4, artist: 5, \dots, exhibition: 50}\}$

$V_{Human}(computer, creative) = \{\text{genius: 2, artist: 5, designer: 12, \dots, inventor: 49}\}$

Finally, we merge the two lists of potential vehicles into one, $V = V_{General} \cup V_{Human}$.

Generation of Slogans

The expression generation process takes the list of vehicle candidates V from metaphor generation process as input, as well as the initial input to the approach, i.e. T and P .

This section is divided as follows. We start by explaining how the semantic and search spaces which the method traverses are constructed. Thereafter, we motivate and define the aspects which we will consider while finding potential solutions, followed by a detailed description of generation algorithm.

Construction of Semantic Spaces From the pool of possible skeletons δ , the approach selects a skeleton s at random. Given a skeleton s , the method constructs two semantic spaces where words in them are used as potential fillers for s . These spaces are (1) interesting I and (2) universal Υ semantic spaces.

The interesting semantic space, which contains words that are favoured, is constructed by obtaining related words, from ω , to the input concept T and a vehicle v from list of vehicle candidates V . The method obtains the k words most strongly related to T . In our case k was empirically set to 150. The method includes related words to v to encourage the generation of metaphorical expressions. For any $v \in V$, the top k related words to v , in ω , are collected while ensuring that they are abstract. This condition is applied because abstraction tends to be required in processing metaphors (Glucksberg 2001). To select only abstract terms, we utilize the abstractness dataset provided by Turney et al. (2011) and keep words with abstractness level ≥ 0.5 . After all related words are obtained, we define I as $\omega(T) \cup \omega(v)$.

We define Υ to be the total semantic space which contains all possible words that could fill s while maintaining its grammatical relations.

The search space of slogans, given a skeleton s , consists of all feasible ways of filling the skeleton with words in I or alternatively in Υ . The task of the expression generator is to traverse the search space and find suitable solutions.

Criteria of good slogans We divide the criteria of good slogans into two categories, filtering and evaluation. Filtering criteria exist to delete any expression that is not acceptable or invalid (boolean), whereas evaluation criteria are employed to be maximised (ratio).

In our method, the filtering criteria are i) relatedness between words within the slogan and ii) positive sentiment. On the other hand, the evaluation criteria consist of i) relatedness to the input, ii) language correctness and word frequencies and iii) figurative devices. Depending on the overall creative goal, different set of evaluation criteria should

be investigated and implemented. For instance, to generate ironic expressions one might use negatively related terms.

Implementation details of these criteria are explained in the remainder of this section, in the Filtering and Evaluation paragraphs.

Algorithm for traversing the search space We employ genetic algorithms to find good slogans in the above detailed space of possible slogans, given a fixed skeleton. We use Deap (Fortin et al. 2012) as the evolutionary computation framework. We use μ to denote the size of the population, G the number of generations to produce, and $Prob_m$ and $Prob_c$ the probability of the mutation and crossover, respectively.

Our algorithm first produces an initial population and then evolves it over a certain number of generations. Starting with the initial population, the employed $(\mu + \lambda)$ evolutionary algorithm produces λ number of offspring by performing multiple crossovers and mutations. The algorithm then puts the current population and offspring through a filtering process (discussed below). The population for the next generation is produced by evaluating the current population and the offspring, and then selecting μ number of individuals. The evolutionary process ends after the specified number of generations.

Initial Population Given the skeleton s , our algorithm begins filling the word (slot) with the most dependent words to it, starting from the root. Using the grammatical relations resource γ , the algorithm ensures that the words satisfy the grammatical relations of s . The algorithm attempts to randomly pick a word residing at the intersection of I and Υ , i.e. interesting and possible. If the intersection is empty, a word is randomly picked from the set of possible fillers Υ . The algorithm repeats the same process for filling the remainder of the words, also taking into account the conditions imposed by the already filled words. However, if the process fails to locate a suitable filler for the next word slot, the whole slogan is discarded and the process starts over. The process continues until the desired number of individual expressions are generated, serving as the initial population.

Given the large knowledge bases used, especially the grammatical relations γ and semantic relatedness ω , it is unlikely for the approach to fail in creating slogans for a given input; however, it is yet possible in some cases such as (1) a rare concept or property with few or noisy associations, (2) a low k threshold or (3) a grammatically incorrect skeleton.

Mutation and Crossover Our algorithm employs only one kind of mutation. The mutation randomly selects and substitutes a word from the expression. In doing so, it follows the same process as was described for the slogan generation for the initial population. Our algorithm applies a one-point crossover. The resultant newly generated child expressions are then put through a grammatical check to verify that all grammatical relations in the expressions exist in our grammatical relations repository γ . A failure of the grammatical check, for any child, results in the disposal of the

child expressions while parent expressions are kept in the population.

Filtering The relatedness model ω is used to check relatedness of words in the slogan against each other. The slogans with unrelated words are filtered out.

The filtering process also removes any expressions with negative sentiments. Advertising slogans tend to contain positive words (Dowling and Kabanoff 1996) which would give the receiver a positive feeling about the brand. As a result, it is essential to employ sentiment analysis in producing slogans. Our filtering process uses the sentiment classifier provided in *Pattern* (Smedt and Daelemans 2012) to classify whether an expression contains any negative words and removes it from the new generation.

The mutation and crossover may produce duplicate slogans or slogans with unrelated words. The filtering stage also takes care of such anomalies. Once a new generation is produced, the filtering process removes any duplicates.

Evaluation In our evaluation metric, we define four main dimensions: i) target relatedness, ii) language correctness, iii) metaphoricity and iv) prosody. Each dimension can be further composed of multiple sub-features. These sub-features are weighted and summed to represent the entire dimension.

Target relatedness measures the relatedness of the words in the slogan to the target input, i.e. T and P , using ω . The relatedness to T and P are two sub-features of the relatedness dimension. The target relatedness is calculated as the mean of the relatedness value of each content word in the expression to the target word.

The language dimension is concerned with how probable is the slogan to be generated with language model ξ . Additionally, another feature which measures how infrequent the individuals are in the slogan, as defined by Özbal, Pighin, and Strapparava (2013).

The metaphoricity dimension contains two sub-features. The first aims at measuring how the words w in the slogan \mathcal{E} are related to both, the tenor T and the vehicle v . This relatedness feature is measured as follows:

$$\text{max_rel}(x) = \operatorname{argmax}_{w \in \mathcal{E}} \omega(w, x) \quad (1a)$$

$$\text{metaphoricity}_1 = \text{max_rel}(T) \cdot \text{max_rel}(v) \quad (1b)$$

The other feature is employed to ensure that there is at least a word that is strongly related to the metaphorical vehicle v but not to tenor T :

$$\text{metaphoricity}_2 = \operatorname{argmax}_{w \in \mathcal{E}} (\omega(w, v) - \omega(w, T)) \quad (2a)$$

The fourth dimension covers four features of prosody: i) rhyme, ii) alliteration, iii) assonance and iv) consonance. The approach makes use of *The CMU Pronouncing Dictionary* (Lenzo 1998) to measure the frequency of repeated sounds between words.

Selection Some of the evaluations involved in our algorithm are conflicting in nature. A normal sorting method for selection, ordering expressions based on the sum of all evaluations, could potentially lead to dominance of one of the evaluations over others, resulting in imbalanced slogans. Therefore, our selection process involves non-dominant sorting algorithm which is more effective when dealing with multiple conflicting objectives (Deb et al. 2002).

Evaluation

We perform two evaluations. The first aims at evaluating the metaphor generation method while the second evaluates the process and the output of the slogan generator method. Future work will address evaluation of the targeted use-case, i.e. a co-creative slogan generator.

In both evaluations, we run crowdsourced surveys on Crowdfunder⁵. These surveys are targeted to the following English speaking countries: United States, United Kingdom, New Zealand, Ireland, Canada, and Australia.

Table 1 lists the concepts and properties defined by us to evaluate the methods. Overall, we had 35 concept-property pairs.

Concept	Properties	Concept	Properties
book	wise, valuable	chocolate	healthy, sweet
computer	creative, mathematical, powerful	painting	creative, majestic, elegant
car	elegant, exotic, luxurious	university	diverse, valuable
coke	sweet, dark	museum	ancient, scientific
love	wild, beautiful, hungry	professor	old, wise, prestigious, smart
newspaper	commercial, international	paper	white, empty, scientific
politician	powerful, dishonest, persuasive, aggressive		

Table 1: List of evaluated input to the system.

Evaluation of Metaphor Generation

The purpose of this evaluation is to find whether using a metaphor interpretation model to select apt vehicles outperforms selecting vehicles solely based on their strong relatedness with the property.

In total, for the inputs defined in Table 1, the method produces 53 vehicles considered apt by the interpretation model, of which 31 are general and 22 human. For each apt vehicle, we select three other vehicles for comparison, as described below. Let $type$ denote the type of the apt vehicle, i.e., $type \in \{General, Human\}$.

1. *Apt*: This is the apt vehicle, in the list V_{type} of vehicles considered apt by the metaphor generation method, for which the following three other vehicles are chosen for comparison.
2. *Strongly related*: a vehicle randomly selected from the vehicle candidates strongly associated with property P (i.e. from top 10% in κ_{type}), but restricted to those that are not considered appropriate by *Meta4meaning* (i.e. not in V_{type}).
3. *Related*: a vehicle associated with property P but not strongly. It is obtained by picking a random vehicle from the bottom 90% of nouns associated with P in κ_{type} .

⁵<http://www.crowdfunder.com>

4. *Random*: a vehicle randomly selected among those nouns that are not associated at all with property P in the knowledge base κ .

Given the 53 apt vehicles, we get 212 metaphors to evaluate overall. We represent each of them as a nominal metaphor of the form “ T is [a/n] v ” (e.g., “computer is an artist”). We then asked judges if the metaphor expresses the intended property (that computer is creative). The judges used a 5-point Likert scale where 1 indicates strong disagreement and 5 strong agreement. The order of metaphors was randomized for each judge. 10 judges were required to evaluate every metaphor.

Evaluation of Slogan Generation

We perform the second evaluation to identify whether the proposed method is capable of producing expressions suitable for the task, i.e. as advertising slogans. A technical sub-goal of the evaluation is also to investigate the effects of the evaluation dimensions of the genetic algorithm on the produced slogans.

Below is how we evaluate the slogan generation method. For every apt vehicle selected in the previous evaluation along with its input, we randomly select two skeletons from the database δ to be filled by the genetic algorithm. We empirically set the following parameters of the genetic algorithm: $\mu = \lambda = 100$, $G = 25$, $Prob_c = 0.4$, $Prob_m = 0.6$.

From the final population produced by the genetic algorithm, we select multiple slogans to be evaluated. We select four slogans which maximize each dimension individually. If possible, we also randomly select a slogan that has a positive value on all four dimensions. Additionally, we select two slogans at random where the slogan has positive values on both the relatedness and language dimensions, and either of the rhetorical dimensions, at least. Lastly, we select the slogan that has the minimum value on all dimensions. As a negative example, some of the above selections might fail because no slogan in the generated population meets the selection criteria. This selection yields 684 slogans to be evaluated. Finally, to present expressions as in a slogan-like style, we detokenize them using *nlTK*⁶ and then capitalise the words in them.

We ask 5 judges to evaluate each selected slogan on a 5-point Likert scale based on the following five aspects: (1) the relatedness of the slogan to the concept and property (i.e. input), (2) the correctness of the language, (3) the metaphoricity, (4) the catchiness, attractiveness and memorability, and (5) the overall appropriation of the expression to be used as a slogan. As phonetic aesthetics can be measured computationally, we instead evaluate the effect of prosody features on the catchiness of the expressions.

Results and analysis

This section presents the results obtained from the evaluations described above.

⁶ <http://www.nltk.org>

Results of Metaphor Generation

Figure 2 is a diverging bar chart illustrating the percentages of judgements on the Likert scale for each type of vehicles. We can observe that apt vehicles performed best. Furthermore, quality drops as relatedness strength weakens.

Overall, judges agreed or strongly agreed 38% of the time that nominal metaphors constructed with apt vehicles expressed the intended property. On the other hand, metaphors where the vehicle was strongly associated with the property (but not apt according to the method) were successful in 28% of the cases. The corresponding agreements are even lower for (non-strongly) related vehicles, 19%, and non-related vehicles, 11%.

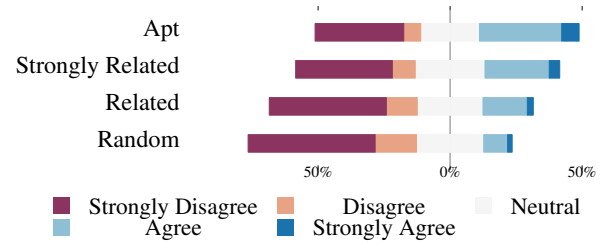


Figure 2: Success of metaphor generation: agreement that the generated metaphor expresses the intended property.

We next consider the means (μ_x) and standard deviations (SD) of the scores in the Likert scale (Table 2). We also provide these statistics for the two vehicle types evaluated (general and human) vehicles. The number of judgements analysed for each of the four selections (Apt, Strongly Related, Related, Random) is 530, where 310 and 220 of them were general and human vehicles, in the same order.

Based on the statistics, we can observe that apt and strongly related human vehicles, retrieved from V_{Human} , received the highest means, 2.98 and 2.57 respectively.

	Apt		Strongly related		Related		Random	
	μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD
General	2.51	1.38	2.45	1.30	2.20	1.25	2.01	1.15
Human	2.98	1.33	2.57	1.31	2.22	1.22	2.00	1.08
Total	2.71	1.38	2.50	1.31	2.21	1.23	2.01	1.12

Table 2: The mean and standard deviation of the judgements of metaphors.

The above results show that there is some difference in favour of apt vehicles. We performed a statistical significance test to examine if it is likely that this difference is due to chance. The null hypothesis is that the scores for apt vehicles and strongly related vehicles come from the same distribution, and any difference is due to random effects; the alternative hypothesis is that the mean for apt vehicles is greater than for strongly related vehicles.

We implemented this test as a permutation test, where the two sets of scores were pooled together and then randomly divided to two sets of the original sizes. We ran one hundred million permutations, obtaining an estimate of the distribution between the means under the null hypothesis.

Based on the test, the p-value is 0.0074. The result suggests that apt vehicles perform statistically significantly better than strongly related vehicles.

Results of Slogan Generation

We analyse the results of slogan generation in this section. Table 3 shows some examples of slogans generated by our method.

Concept	Property	Vehicle	Output
computer	creative	artist	Talent, Skill And Support.
			Follow Questions. Start Support.
		poet	Work Unsupervised.
			Younger Than Browser.
car	elegant	dancer	The Cars Of Stage.
painting	creative	literature	You Ca N't Sell The Fine Furniture.
politician	persuasive	orator	Excellent By Party. Speech By Talent.
	dishonest	thief	Free Speech.
	aggressive	predator	Media For A Potential Attack.

Table 3: Selected examples of generated slogans by the proposed method.

In the following analysis, we consider an individual slogan successful, if the mean score for its overall suitability (the 5th question in the evaluation questionnaire) is above 3. On average, 35% of generated slogans were considered suitable. The input with most suitable slogans was *computer-powerful*, with 13 suitable slogans out of 20. On the other hand, the input *newspaper-international* had the least number of good slogans, 1 out of 12. This analysis shows that the method has successfully generated at least one suitable slogan for each input. Given that the method actually generates an entire population of slogans, more options would be available for an actual user to select from.

Table 4 shows the mean μ_x and standard deviation SD for all slogans evaluated, grouped by the selection methods described in the Evaluation of Slogan Generation section. Letters in the Selection column reflect the four dimensions in the genetic algorithm, i.e. (*r*)elatedness to input, (*l*)anguage, (*m*)etaphoricity, and (*p*)rosody. $pos(*)$ denotes a positive value on all mentioned dimensions only, whereas $min(*)$ and $max(*)$ ensures that they are minimised and maximised, respectively. The number of slogans evaluated for each group is expressed as n .

Selection	n	Relatedness		Language		Metaphoricity		Catchyness		Overall	
		μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD
$pos(r, l, m, p)$	262	3.05	0.69	3.15	0.67	2.91	0.60	2.98	0.67	2.92	0.68
$pos(r, l, m)$	93	3.01	0.76	3.06	0.72	2.93	0.61	2.93	0.71	2.87	0.70
$pos(r, l, p)$	111	3.00	0.73	3.17	0.63	2.91	0.63	2.88	0.59	2.86	0.66
$max(r)$	100	3.11	0.70	3.19	0.66	2.90	0.61	2.95	0.68	2.90	0.70
$max(l)$	105	2.89	0.70	3.16	0.70	2.83	0.59	2.91	0.65	2.80	0.68
$max(m)$	88	2.94	0.73	3.01	0.64	2.90	0.62	2.91	0.66	2.83	0.67
$max(p)$	96	2.93	0.76	3.11	0.71	2.91	0.68	2.86	0.67	2.83	0.69
$min(r, l, m, p)$	104	2.77	0.69	2.98	0.65	2.78	0.65	2.82	0.65	2.75	0.70

Table 4: Mean and standard deviation of various judgements of slogans grouped by different selections.

Observing the overall suitability among all selections, we notice that slogans with balanced dimensions, i.e. $pos(*)$, were appreciated more than slogans with a dominant, $max(*)$, dimension.

Correctness of the language used in slogans received the highest average rating overall. This is mostly because the

language of slogans is checked throughout the entire method (e.g. filling skeletons, mutation, and crossover).

From the examples in Table 3 and opinions on the metaphoricity of generated slogans (Table 4), we can see that the method is capable of generating rhetorical expressions.

Individually maximised dimensions seem to have some correspondence to judgements of their relevant question. For instance, slogans maximising the relatedness dimension, $max(r)$, were judged to be related to the input considerably higher than other selections.

Finally, slogans that had the lowest evaluation values on the four dimensions have also received the lowest agreements on all five questions.

We also perform permutation tests on judgements obtained on generated slogans regarding their overall suitability. In this analysis, we divide the data into three sets based on the selection mechanism (i.e. slogans with *balanced* dimensions, slogans with a *maximised* dimension and slogans with *least* evaluation scores). Using one hundred million permutations, we compare the means under the following alternative hypotheses:

1. $\mu_x(\text{balanced}) > \mu_x(\text{maximised})$
2. $\mu_x(\text{balanced}) > \mu_x(\text{least})$
3. $\mu_x(\text{maximised}) > \mu_x(\text{least})$

Among the tests, only in the second case is the null hypothesis rejected, with a p-value of 0.0286.

These statistics confirm that slogans with balanced values on multiple dimensions (i.e. related to the input, grammatically correct, and have at least one rhetorical device) improve the suitability of slogans.

Discussion and Conclusions

In this paper, we have described automatic methods for generating first metaphors and then slogans. Also, we have evaluated both steps individually by crowdsourcing questionnaires.

The metaphor generation method employs a metaphor interpretation model –*Meta4meaning*– to measure the aptness of vehicle candidates. We have evaluated the method against metaphors generated based on strong relatedness to input property. The results of the evaluation indicate that using a metaphor interpretation model produces better metaphors.

Nevertheless, as the metaphor generation method relies mainly on *Meta4meaning*, a failure of interpreting a metaphor by the model for any of its limitations, c.f. Xiao et al. (2016), might treat apt vehicles as non-apt.

Our method for generating slogans is based on genetic algorithms using multi-objective selection. The method has successfully created slogans that were considered suitable, related, grammatically correct, metaphorical and catchy, based on crowdsourced opinions.

A possible future direction for metaphor generation is to combine an interpretation model with additional measurements to reach aptness scores matching how humans perceive metaphors.

Studying the effects of adjusting the parameters of the methods on the results is left for future work. These parameters could be altered dynamically based on the interactions between the user and the system, which would motivate collaborations between humans and computers in solving creative tasks. Finally, the proposed method could be compared to human-made slogans for the same tasks or evaluated in other domains (e.g. creating news titles) with appropriate adaptations.

Acknowledgements

This work has been supported by the Academy of Finland under grant 276897 (CLiC).

References

- Alnajjar, K.; Hämäläinen, M.; Chen, H.; and Toivonen, H. 2017. Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the Eighth International Conference on Computational Creativity (ICCC 2017)*, 25–32. Atlanta, United States: Georgia Institute of Technology.
- Burgers, C.; Konijn, E. A.; Steen, G. J.; and Iepsma, M. A. R. 2015. Making ads less complex, yet more creative and persuasive: The effects of conventional metaphors and irony in print advertising. *International Journal of Advertising* 34(3):515–532.
- Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6(2):182–197.
- Dowling, G. R., and Kabanoff, B. 1996. Computer-aided content analysis: What do 240 advertising slogans have in common? *Marketing Letters* 7(1):63–75.
- Evert, S. 2008. Corpora and collocations. In Lüdeling, A., and Kytö, M., eds., *Corpus Linguistics. An International Handbook*, volume 2. Berlin: Mouton de Gruyter. 1212–1248.
- Fortin, F.-A.; De Rainville, F.-M.; Gardner, M.-A.; Parizeau, M.; and Gagné, C. 2012. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* 13:2171–2175.
- Galvan, P.; Francisco, V.; Hervás, R.; Mandez, G.; and Gervás, P. 2016. Exploring the role of word associations in the construction of rhetorical figures. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.
- Giora, R. 2003. *On our mind: Salience, context, and figurative language*. Oxford University Press.
- Glucksberg, S. 2001. *Understanding Figurative Language: From Metaphor to Idioms*. Oxford University Press.
- Harmon, S. 2015. FIGURE8: A novel system for generating and evaluating figurative language. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, 71–77. Park City, Utah: Brigham Young University.
- Lenzo, K. 1998. The cmu pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
- McGregor, S.; Agres, K.; Purver, M.; and Wiggins, G. 2015. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *Journal of Artificial General Intelligence* 6(1):55–86.
- Miller, D. W., and Toman, M. 2014. An analysis of rhetorical figures and other linguistic devices in corporation brand slogans. *Journal of Marketing Communications* 7266(July 2015):1–20.
- Özbal, G.; Pighin, D.; and Strapparava, C. 2013. Brainsup: Brainstorming support for creative sentence generation. In *ACL (1)*, 1446–1455.
- Reece, B. B.; Van den Bergh, B. G.; and Li, H. 1994. What makes a slogan memorable and who remembers it. *Journal of Current Issues & Research in Advertising* 16(2):41–57.
- Richards, I. A. 1936. *The Philosophy of Rhetoric*. London: Oxford University Press.
- Smedt, T. D., and Daelemans, W. 2012. Pattern for python. *Journal of Machine Learning Research* 13(Jun):2063–2067.
- Strapparava, C.; Valitutti, A.; and Stock, O. 2007. Automating two creative functions for advertising. In Cardoso, A., and Wiggins, G., eds., *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 99–108. London, UK: Goldsmiths, University of London.
- Tom, G., and Eves, A. 1999. The Use of Rhetorical Devices in Advertising. *Journal of Advertising Research* 39(4):39–43.
- Tomašić, P.; Znidaršič, M.; and Papa, G. 2014. Implementation of a slogan generator. In *Proceedings of 5th International Conference on Computational Creativity, Ljubljana, Slovenia*, volume 301, 340–343.
- Turney, P. D.; Neuman, Y.; Assaf, D.; and Cohen, Y. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, 680–690.
- Veale, T., and Li, G. 2013. Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 660–670.
- Xiao, P., and Blat, J. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity (ICCC 2013)*, 8–15. Sydney, Australia: The University of Sydney.
- Xiao, P.; Alnajjar, K.; Granroth-Wilding, M.; Agres, K.; and Toivonen, H. 2016. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.