



# EMBEDDIA

## Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action

Call: H2020-ICT-2018-1

Call topic: ICT-29-2018 A multilingual Next generation Internet

Project start: 1 January 2019

Project duration: 36 months

### D5.6: Creative multilingual technology for news and headline generation (T5.3)

#### Executive summary

This deliverable describes methods, developed in T5.3, for generating creative language and news headlines. First, we describe methods for producing metaphors and metaphorical expressions, methods for introducing creativity into multilingual news headlines, and methods for turning existing headlines into humorous ones. Next, we present an approach to creative news article generation from raw data using a novel creative architecture. Finally, we explore headline generation in a multilingual low-resource setting using encoder-decoder neural architectures.

Partner in charge: JSI

#### Project co-funded by the European Commission within Horizon 2020 Dissemination Level

| PU | Public  | PU |
|----|---|----|
| PP | Restricted to other programme participants (including the Commission Services)        | –  |
| RE | Restricted to a group specified by the Consortium (including the Commission Services) | –  |
| CO | Confidential, only for members of the Consortium (including the Commission Services)  | –  |



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

## Deliverable Information

| Document administrative information |  |
|-------------------------------------|--|
| Project acronym:                    | <b>EMBEDDIA</b>  |
| Project number:                     | <b>825153</b>  |
| Deliverable number:                 | <b>D5.6</b>  |
| Deliverable full title:             | <b>Creative multilingual technology for news and headline generation</b> |
| Deliverable short title:            | <b>Creative news generation</b>  |
| Document identifier:                | <b>EMBEDDIA-D56-CreativeNewsGeneration-T53-submitted</b>                 |
| Lead partner short name:            | <b>JSI</b>   |
| Report version:                     | <b>submitted</b>   |
| Report submission date:             | <b>31/10/2021</b>  |
| Dissemination level:                | <b>PU</b>  |
| Nature:                             | <b>R = Report</b>  |
| Lead author(s):                     | <b>Matej Martinc (JSI), Khalid Alnajjar (UH)</b>                         |
| Co-author(s):                       | <b>Matthew Purver (QMUL), Anita Valmarska (JSI)</b>                      |
| Status:                             | <b><u>_</u> draft, <u>_</u> final, <u>x</u> submitted</b>                |

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

## Change log

| Date       | Version number | Author/Editor             | Summary of changes made  |
|------------|----------------|---------------------------|--|
| 20/12/2020 | v0.1           | Anita Valmarska (JSI)     | Report structure.  |
| 07/01/2021 | v0.2           | Khalid Alnajjar (UH)      | Written section about creative language generation of headlines and slogans.   |
| 28/09/2021 | v0.3           | Matej Martinc (JSI)       | Written section about headline generation using sequence to sequence models.   |
| 30/09/2021 | v0.4           | Matthew Purver (QMUL)     | Written section about creative architecture for article generation.  |
| 30/09/2021 | v0.5           | Matej Martinc (JSI)       | Written introduction and conclusion sections, changed the section structure and proofreading.  |
| 03/10/2021 | v0.6           | Marko Robnik-Šikonja (UL) | Internal review.   |
| 12/10/2021 | v0.7           | Marko Pranjić (TRI)       | Internal review.   |
| 13/10/2021 | v0.8           | Khalid Alnajjar (UH)      | Implemented suggestions from internal review in section about creative language generation of headlines and slogans.                                     |
| 13/10/2021 | v0.9           | Matthew Purver (QMUL)     | Implemented suggestions from internal review in section about creative architecture for article generation.  |
| 15/10/2021 | v0.10          | Matej Martinc (JSI)       | Implemented suggestions from internal review in the introduction, conclusion and in section about headline generation using sequence to sequence models. |
| 20/10/2021 | v0.11          | Nada Lavrač (JSI)         | Quality control.   |
| 26/10/2021 | final          | Matej Martinc (JSI)       | Report finalized.  |
| 29/10/2021 | submitted      | Tina Anžič (JSI)          | Report submitted.  |

## Table of Contents

|  |    |
|--|----|
| 1. Introduction.....   | 5  |
| 2. Creative Generation of Headlines and Slogans .....  | 6  |
| 2.1 Generation of News and Computational Creativity .....  | 6  |
| 2.2 Generation of Metaphorical Expressions .....   | 7  |
| 2.3 Making Headlines Humorous .....  | 10 |
| 3. A Creative Architecture for Article Generation .....  | 13 |
| 4. Headline Generation Using Sequence to Sequence Models.....  | 14 |
| 4.1 Methodology .....  | 15 |
| 4.2 Experiments .....  | 18 |
| 5. Conclusions and further work.....   | 21 |
| 6. Associated outputs .....  | 21 |
| References .....   | 23 |
| Appendix A: No time like the present: methods for generating colourful and factual multilingual news headlines.. | 27 |
| Appendix B: Computational generation of slogans .....  | 35 |
| Appendix C: When a Computer Cracks a Joke: Automated Generation of Humorous Headlines.....                       | 68 |
| Appendix D: Creative Language Generation in a Society of Engagement and Reflection .....                         | 76 |
| Appendix E: Parsing Text in a Workspace for Language Generation.....   | 80 |
| Appendix F: Evaluating Natural Language Descriptions Generated in a Workspace-Based Architecture.....            | 94 |

## List of abbreviations

|         |   |
|---------|---|
| DoA     | Description of Action                                   |
| EC      | European Commission                                     |
| BERT    | Bidirectional Encoder Representations from Transformers |
| BERT-ED | BERT encoder-decoder                                    |
| ExM     | Ekspress Meedia   |
| GA      | Grant Agreement   |
| JSI     | Jožef Stefan Institute                                  |
| UH      | University of Helsinki                                  |
| QMUL    | Queen Mary University of London                         |
| NLG     | Natural Language Generation                             |
| SS      | Semantic Similarity                                     |
| CD      | Cosine Distance   |
| T       | Task  |
| WP      | Work Package  |

# 1 Introduction

The EMBEDDIA work package 5 (WP5) deals with Natural Language Generation (NLG) and aims to develop news automation systems transferable across languages and domains. It consists of three main tasks, namely T5.1, which deals with text generation from structured data and adaptation of NLG technology to media house environment, T5.2, which deals with multilingual storytelling and dynamic content generation, and T5.3, the results of which are described in this deliverable and deals with development of tools for creation of figurative language and headlines in a multilingual environment. More specifically, in the scope of this task, we investigate methods for automatic creation of creative expressions that will make the text more varied, methods for automatic generation of cross-cultural figurative expressions, and methods for automatic generation of creative and factual news headlines. These methods should also be adapted to work in a multi- or cross-lingual environment. The above described problems can be tackled in several ways, therefore within T5.3 we implemented three distinct approaches, each of them employing a different technique:

- The first option for automatic generation of figurative language and headlines is to inject creative expressions into pre-existing headlines, which were either generated by automated journalism system or written by journalists. Here, we employ techniques from the field of computational creativity, which is, based on the definition of Colton & Wiggins (2012), “the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative”. Creative Natural Language Generation (NLG) is a computational creativity topic focusing on automatically producing creative text such as stories, poems, and slogans. To demonstrate the applicability of the proposed approach for generation of creative expressions, we briefly describe our study on slogan generation (Alnajjar & Toivonen, 2021). Creative NLG has received a lot of research interest in the past. However, the main goal of such systems has seldom been to communicate factual information in a creative way, as journalist try to do when inventing catchy headlines. Therefore, the task of headline generation, which we tackle with computational creativity methods, is distinctive from the current state of the field in its pursuit of more than just creativity. In our work, we approached the topic from an interdisciplinary standpoint.

Since one of the purposes of the EMBEDDIA project is to develop tools and methods that facilitate provision of multilingual resources for several less-resourced European languages, we tackled automatic creation of figurative language through cross-lingual word embeddings. The proposed method allows to leverage Natural Language Generation (NLG) techniques in a modular way, by separating the message construction phase from the surface realization phase (i.e. realizing the semantic meaning into a sentence). Thus, a single NLG method would be capable of producing multilingual natural text (e.g., news articles) serving a wide audience speaking different languages. The proposed method imitates journalists in the use of creative language (such as metaphors, proverbs and humour) to write news articles that are engaging and interesting to the audience. In contrast, most current automatically generated news articles tend to focus on constructing factual messages using a fixed set of constrained human-authored templates, which makes text appear repetitive and dull to the audience. For this reason, we research computational creativity methods for making generated text more colourful and creative.

- The second approach to natural language generation also employs computational creativity techniques but does not focus on generation of short texts but rather on generation of entire articles. This approach generates articles from raw data, instead of injecting creative language into the existing text. The advantage is that in order to produce an article, one only requires a set of basic conceptual frames describing the concepts and relations between them as an input. From this input, the article is generated in a cyclic process with two main phases, engagement and reflection, during which representations are generated and evaluated, respectively, in an iterative fashion.
- The third approach does not leverage methods from computational creativity but relies on conventional language generation methods. The headline generation is approached from the informational aspect of a headline. If we consider a headline as a vehicle that carries the most important

information about the event or topic described in the news article, the headline shall be a sort of summary. Recent automatic summarization approaches, such as BART (Lewis et al., 2020), employ an encoder-decoder transformer architecture (Vaswani et al., 2017), which ‘translates’ the input text into an output summary. The encoder-decoder approach to headline generation was used in the past (see Section 4) and in the NLP community there is a consent that encoder-decoder architecture is suitable for a variety of text generation tasks. This technology is still rarely used in the production due to limitations, such as a special case of overfitting called ‘hallucination’, bad interpretability, and lack of resources, since these approaches require vast amounts of text for successful training. Nevertheless, due to its potential, we decided to test the encoder-decoder architecture. We focused mostly on the problem of how to reduce the amount of required text for successful training and explored several techniques for reducing the amount of needed data, in order to adapt the approach to low-resource scenarios and make it transferable to less-resourced languages. We test the approach in a multi-lingual setting and obtain promising results.

To summarize, the main contributions of this deliverable are the following:

- A novel method for generation of headlines and creative expressions that relies on methods from the field of computational creativity and generates creative language by injecting creative expressions into pre-existing headlines. The approach is described in Section 2.
- A novel method for generation of news articles from raw data in a cyclic process with two main phases, engagement and reflection. The approach is described in Section 3.
- A novel method for generation of news headlines in a multilingual low-resource environment, using the neural encoder-decoder architecture. The approach is described in Section 4.

After presentation of the three approaches, the deliverable presents conclusions and related work in Section 5. The final Section 6 contains the associated outputs of the work done within T5.3. All articles published as part of the task are enclosed in the appendices.

## 2 Creative Generation of Headlines and Slogans

This section is divided into three part. In Section 2.1, we present a novel method for injecting creative expressions into news headlines, Section 2.3 describes a method for generation of creative expressions and the employment of the method for slogan generation, and Section 2.3 presents a computationally creative approach to generating humorous versions of existing headlines.

### 2.1 Generation of News and Computational Creativity

Existing methods for automated news generation typically fill linguistic templates written by journalists with suitable blocks of information from structured data sources. While template-based methods give strong control over the generated output and ensure correctness of information, they tend to be repetitive and look mechanistic. We have sought to make computer-generated news more varied and colourful by introducing creative expressions in news headlines (Alnajjar et al., 2019).

Headlines must relate to the news articles and briefly describe them while motivating readers to visit the website and read the article. In order to make automatically generated headlines more colourful, we developed algorithmic approaches that extend creative expressions in headlines produced by the news generation system described in other WP5 tasks. The news generation system focuses on producing descriptive and factual news and headlines; while they are informative, they may be dull to their readers. Therefore, the main focus of this work is to introduce creativity to such automatically generated headlines by 1) prepending a well-known and related expression to the headline and 2) inserting metaphorical expressions.

We have experimented with two approaches, inspired by previous research on generating figurative language (Veale & Li, 2013a; Alnajjar et al., 2017) and creative headlines (Lynch, 2015; Gatti et al., 2015), to add a creative touch to news headlines generated by the automated journalism system. The methods operate in a multilingual setting by using cross-lingual word embeddings. With cross-lingual word embeddings we transfer knowledge from English, with rich linguistic resources, into a less-resourced language, e.g., Finnish. As a result, our methods compose creative language in multilingual settings.

*Phrase-copying* is the first method and presents a suitable well-known phrase (e.g., movie title) to readers as a catchy title (i.e., it draws attention) along with the factual message (c.f. Table 1 in Appendix A for examples). The output is structured as “*phrase: headline*”. The idea here is that juxtapositioning the phrase with the headline will catch the attention of readers and motivate them to click on the headline to read the news article, while keeping the factual content of the original headline intact. Two types of well-known phrases were used in our method, namely 1) proverbs in each language and 2) movie titles. The matching of well-known expressions to headlines takes into account the semantic similarity and relatedness, and prosody. Semantic similarity is used for coherence of the resulting combination, while prosody is intended to increase catchiness of the result.

The second method *figurative-injection* injects figurative phrases (e.g., similes and metaphors) into headlines, depending on the polarity of the news. If the given headline has polarity with respect to the main entity in the headline, a political party or candidate in our case, then the method adds a figurative comparison that is stereotypically associated with the polarity. The aim is that this comparison indirectly attributes properties to the entity of the headline, thereby emphasizing the polarity in a creative, figurative way.

We have evaluated the methods by running a crowdsourced evaluation asking online judges to evaluate both the baseline (non-creative) headlines originally produced by a NLG tool and the modified (creative) headlines by the two methods described above. They were then asked to evaluate the headline on a 5-point Likert scale against the following claims:

1. The headline is descriptive of the article.
2. The headline is grammatically correct.
3. The headline is catchy.
4. The headline is creative.
5. The headline can be considered offensive.
6. The headline is generated by a computer.

The evaluation process resulted in 3,000 unique judgments from crowdsourcing, 1,000 for each type of headlines and 10 judges per headline. The score of each headline on the above questions is represented by the mean judgments received on the Likert scale. Table 1 shows the mean and standard deviation of judgments received on each question for the three types of headlines, and Figure 1 gives the diverging bar charts for the answers.

The results suggest that our system is capable of making news headlines more creative and catchier while retaining the original meaning of the headline. This is in line with the requirements set for the task in involving creativity in the context of factual text generation.

***This work is described in full in Alnajjar et al. (2019), attached here as Appendix A.***

## 2.2 Generation of Metaphorical Expressions

Slogans are concise advertising messages that aim to catch the attention of the audience and increase the recall of the product or brand. We proposed a novel method for automatically generating slogans (Alnajjar & Toivonen, 2021), given a target concept (e.g., car) and an adjectival property to express (e.g., elegant) as input. A key component in our approach is a novel method for generating nominal metaphors,

**Table 1:** The mean  $\mu_x$  and standard deviation  $SD$  of judgments received for each type of generated headlines on the six questions. The best result for each question appears in boldface.

\* The value is statistically significantly different ( $p < 0.05$ ) from the value for the baseline headline (non-parametric permutation test with one hundred million repetitions, one-tailed, not corrected for multiple testing).

|                      | Descriptive |      | Grammatical |      | Catchy      |      | Creative     |      | Offensive   |      | Comp.gen.   |      |
|----------------------|-------------|------|-------------|------|-------------|------|--------------|------|-------------|------|-------------|------|
|                      | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ | $\mu_x$      | $SD$ | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ |
| <i>Baseline</i>      | <b>3.91</b> | 0.87 | <b>3.80</b> | 0.86 | 3.31        | 0.93 | 3.18         | 0.94 | <b>2.46</b> | 1.13 | <b>2.87</b> | 1.01 |
| Phrase-copying       | 3.75*       | 0.93 | 3.71*       | 0.88 | <b>3.35</b> | 0.95 | <b>3.35*</b> | 0.93 | 2.61*       | 1.12 | 2.97*       | 1.01 |
| Figurative-injection | 3.70*       | 0.95 | 3.65*       | 0.91 | <b>3.35</b> | 0.93 | 3.33*        | 0.95 | 2.83*       | 1.15 | 3.04*       | 1.00 |

using a metaphor interpretation model, to allow generating metaphorical slogans. Since slogans are structurally very similar to headlines, i.e. being short and catchy, yet offering representative descriptions, the outcomes of this research lead us to integrate colorful headline generation with metaphor generation.

In our metaphor generation component, we used the successful word-embeddings-based method for interpreting nominal metaphors (Xiao et al., 2016). Our method does not use an existing word embeddings model such as Word2Vec or FastText, but builds new word embeddings based on word associations on the paradigmatic level for calculating metaphorical scores. This is needed since off-the-shelf models do not capture the metaphoric nuances in the data, but rather produce a more semantically representative view of the language. For a given input, the metaphor generation component retrieves potential metaphorical candidates for conveying the desired adjectival property from existing knowledge-bases of nouns and their stereotypical adjectival associations (Veale & Li, 2013b; Alnajjar et al., 2017). The metaphorical interpretability of the candidates is then predicted using the metaphorical interpretation model. High-ranked candidates predicted to highlight the desired property are passed to the slogan generation component, where they are utilized in assessing the metaphoricity of the slogans as a part of the generative process.

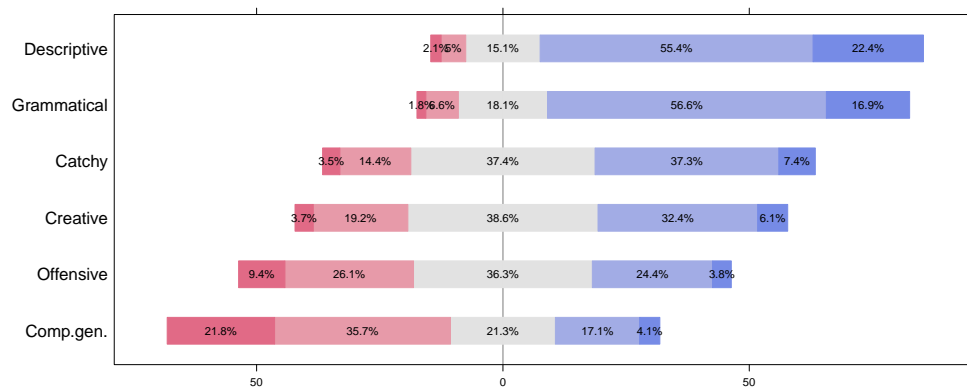
The method for generating slogans extracts skeletons (a parse tree of a sentence where content words are replaced with a placeholder, and where grammatical relations between words and part-of-speech tags are maintained) from existing slogans. It then fills a skeleton in with suitable words by utilizing multiple linguistic resources (such as a repository of grammatical relations, and semantic and language models) and genetic algorithms to optimize multiple objectives such as semantic relatedness, language correctness and usage of rhetorical devices.

Briefly, the genetic algorithm constructs an initial population, which then goes through an evolutionary process (mutation and crossover) for a fixed number of iterations. During each iteration, the fittest individuals, whether they are in the existing population or result from the evolutionary process, survive to the next iteration.

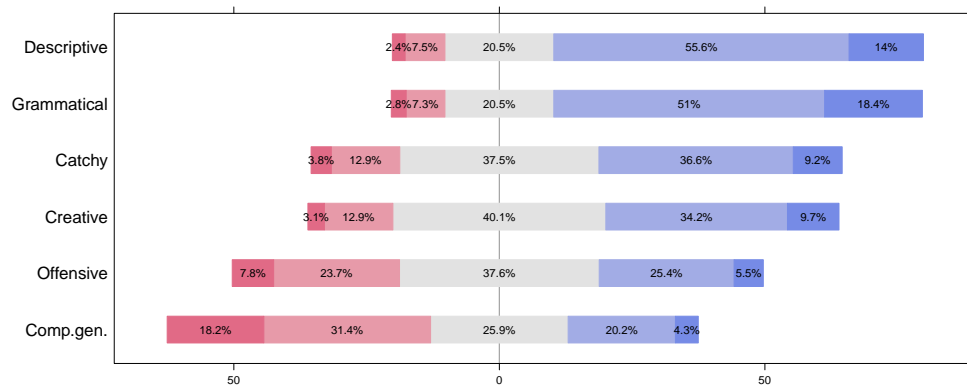
In our genetic algorithm, the initial population consists of skeleton copies filled with content words that are related to the input. During the filling stage, grammatical constraints are checked to ensure correct language. When an individual is mutated, a random content word is selected and replaced with a placeholder, which is then filled by a new relevant word. The crossover selects a single point in two slogans and swaps the preceeding words between them. We defined four components of fitness function to be maximised with the genetic algorithm, 1) relatedness to the input, 2) language correctness, 3) metaphoricity, and 4) prosody. Every component is composed of multiple sub-functions that measure its aspects.

The relatedness has two sub-functions, one for measuring the semantic relatedness to the input concept and the other for calculating the same measurement but to the input property. Language correctness looks at the probability of the slogan to be generated using a language model and the infrequency

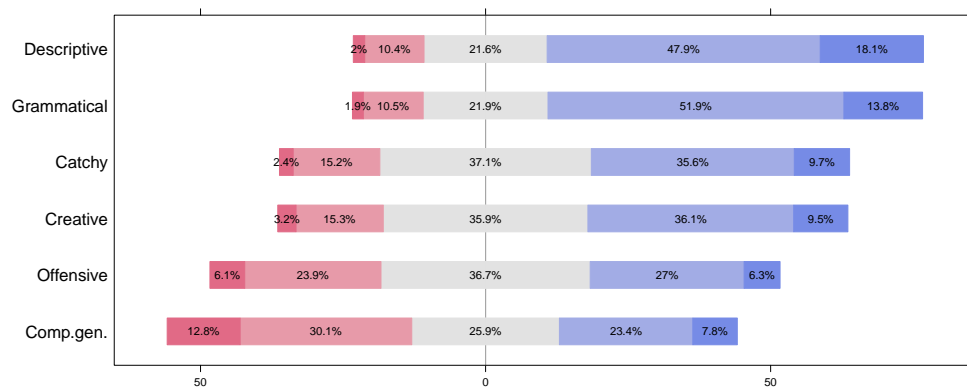




(a) Baseline



(b) Phrase-copying



(c) Figurative-injection

■ Strongly Disagree    ■ Disagree    ■ Neutral  
■ Agree    ■ Strongly Agree

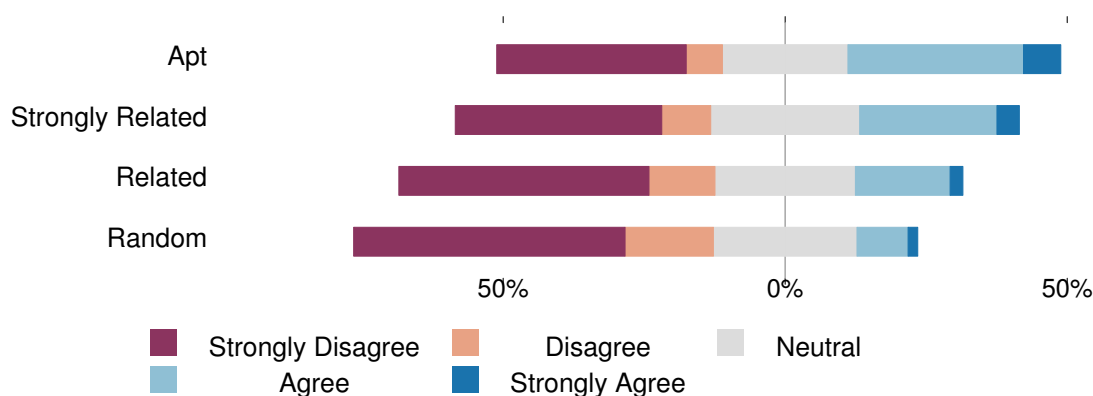
**Figure 1:** Diverging bar charts illustrating the percentage of judgments received on each question for the three types of headlines.

of the chosen words as an approximation of surprise. We proposed a novel measure to assess the metaphoricity of an expression. The measure includes two sub-components; the first ensures that the expression contains words related to the input and metaphorical concepts, while the second measures

and encourages metaphoricity arising from words related to the metaphorical concept but *not* to the input concept. Lastly, the prosody component returns a value based on the rhyme, assonance, alliteration and consonance found in the slogan.

We evaluate the metaphor and slogan generation methods by running crowdsourced surveys. On a 5-point Likert scale, we ask online judges to evaluate whether the generated metaphors, along with three other metaphors generated using different methods, highlight the intended property. The slogan generation method is evaluated by asking judges to rate generated slogans from five perspectives: (1) how well is the slogan related to the topic, (2) how correct is the language of the slogan, (3) how metaphoric is the slogan, (4) how catchy, attractive and memorable is it, and (5) how good is the slogan overall. Similarly, we evaluate existing expert-made slogans.

Figure 2 shows the diverging bar chart of answers returned by human judges on the Likert scale for the metaphor generation evaluation. In the figure, apt metaphors are metaphors generated by our method, whereas the remaining variants are randomly picked based on the association strength between the metaphorical concept and the desired property. The empirical results indicate a clear preference towards metaphors produced by our metaphor generation method that it is capable of producing apt metaphors. The metaphor interpretation model has also been utilized successfully by Hämäläinen & Alnajjar (2019) to measure and generate Finnish metaphors, which implies the successful transfer of our metaphor generation method to other languages.



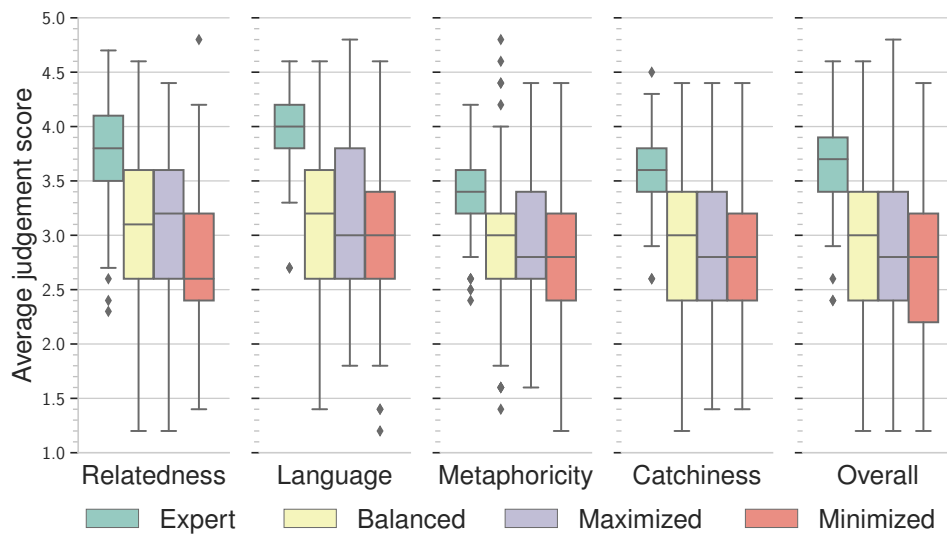
**Figure 2:** Success of metaphor generation: agreement that the generated metaphor expresses the intended property.

Regarding the results on our slogan generator, Figure 3 provides a general summary of the performance of our method and how it compares to slogans written by experts. In the figure, distributions of answers collected during the evaluation are shown. To assess the effect of the defined internal dimensions in our method, we put slogans into three groups depending on their scores on the four dimensions. From the results, we can see that slogans with balanced dimensions (i.e. the four dimensions are taken into account and have a positive value) outperform maximizing a single dimension, overall. This suggests that optimizing our defined dimensions does indeed affect the quality of slogans positively. Furthermore, based on our analysis of the results (c.f. Appendix B for more details), the method has successfully produced at least one effective slogan for every evaluated input.

*This work is described in full in Alnajjar & Toivonen (2021), attached here as Appendix B.*

## 2.3 Making Headlines Humorous

Automated news generation has become a major interest for news agencies. Often headlines for such automatically generated news articles are unimaginative as they have been generated with ready-made templates. We present a computationally creative approach for headline generation that can generate humorous versions of existing headlines (Alnajjar & Hämäläinen, 2021). The headlines still need to be topical, but we want them to provoke the readers' interest by the means of humor. Our method uses



**Figure 3:** Distributions of mean judgements of slogans, for expert-written as well generated ones with different selection methods (balanced, maximized or minimized internal dimensions). Results are given separately for different human judgments (relatedness, language, metaphoricity, catchiness and overall quality). For each judgment, the “maximized” results shown are for the case where the corresponding internal evaluation dimension was maximized by the method; the “overall” case is their aggregation. Plots indicate the median, 1st and 3rd quartiles and 95% intervals.

word embeddings in a novel way to measure two of the important humor characteristics: surprise and coherence. Only surprise can provoke a humor reaction, but coherence assures that a joke will be understood.

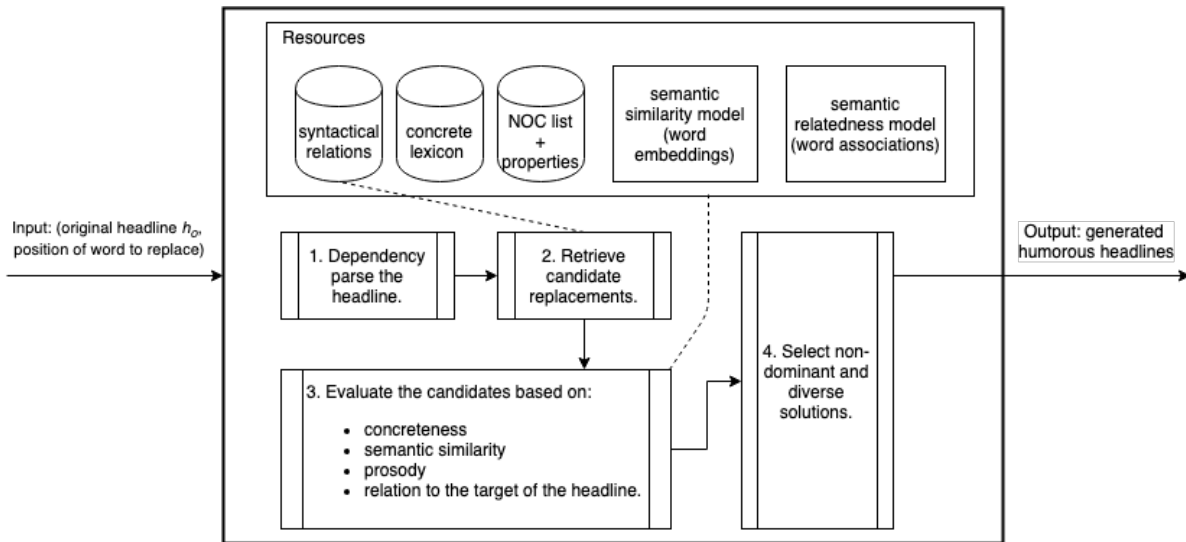
Our system takes an existing headline from the corpus of altered headlines (Hossain et al., 2019) (Humicroedit dataset). This corpus has been syntactically parsed by us, and it has been tagged for the words that should be replaced by its original authors. For a selected headline, our system tries to find suitable humorous replacement words.

We assess the different potential humorous replacements in terms of multiple parameters: prosody, concreteness, semantic similarity of the replacement to the original word, and the semantic relatedness of the replacement to negative words describing the target. An overall view of our method is depicted in Figure 4.

To evaluate our method, we randomly selected 83 headlines from the Humicroedit dataset that can be altered by our method. For each headline, we sent it to our method to generate humorous alternatives, ranked by the non-dominant sorting, out of which we randomly selected three to be evaluated from the top humorous headlines. We asked five people on a crowd-sourcing platform to rate the headlines based on the following questions:

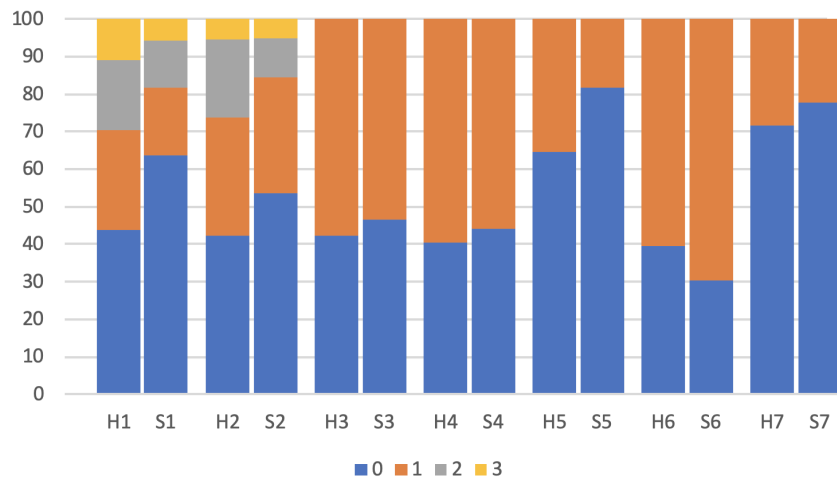
1. The altered headline is humorous.
2. The altered word is surprising.
3. The altered word fits into the headline.
4. The altered word is concrete.
5. The joke of the headline makes fun of a person or a group of people (also known as the target of the joke).
6. The altered word shows the target in a negative light.
7. The altered word is a pun of the original word.

We evaluate the first two questions on the scale form 0 to 3 (0-*Not funny*, 1-*Slightly funny*, 2-*Moderately funny* and 3-*Funny*, following the evaluation conducted in Hossain et al. (2017)). In the second question, we replace the word funny with surprising in the answer alternatives. The rest of the questions are yes/no questions. The sixth question is only visible if the fifth question has been answered affirmatively.



**Figure 4:** A diagram visualizing the process of humor generation.

Figure 5 shows the result for each question using the human evaluation. The results for the human edited titles (H) and the ones produced by our method (S) are shown side by side. From the question 3 onward, 0 marks negative and 1 affirmative answer. Our system scored slightly lower than humans, which is to be expected due to the difficulty of the problem.



**Figure 5:** Distribution of answers for the evaluation questions. H marks human authored headlines, and S computer authored ones.

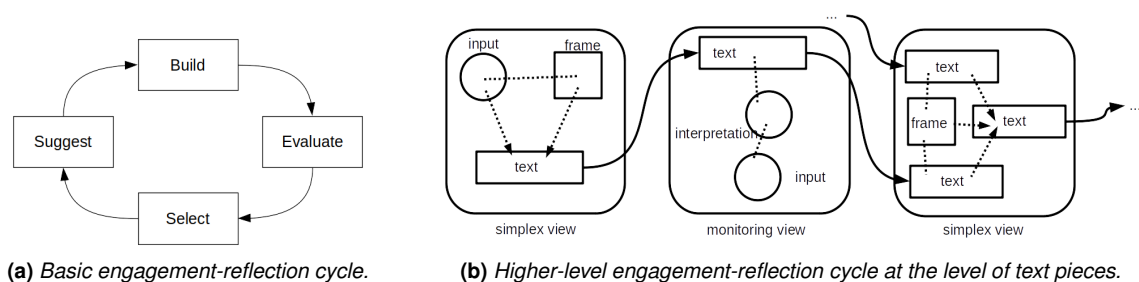
In terms of humor, our system managed to produce at least slightly humorous headlines 36% of the time, whereas people produced at least slightly humorous headlines 56% of the time. Overall, our system has achieved good humor scores in human evaluation and high fidelity to the original message of news headlines.

*This work is described in full in Alnajjar & Härmäläinen (2021), attached here as Appendix C.*

### 3 A Creative Architecture for Article Generation

Another focus in the area of creative generation for news has been the generation of articles from raw data. In contrast to the work in Sections 2 and 4, the task here is to generate complete articles, rather than smaller spans like headlines or metaphorical phrases; and in contrast to the article generation work in Task T5.2 and Deliverable D5.5, we use a computational creativity architecture rather than more conventional natural language generation methods. As a result, the work is more exploratory and currently more restricted in domain: here, we implement and evaluate the approach on the domain of generating weather reports.

The architecture is based in the *engagement-reflection (E-R)* cycle model (Sharples, 1998) used in many approaches to computational creativity (see e.g., Pérez y Pérez & Sharples, 2001; Pérez y Pérez et al., 2013). In the E-R model, the overall process iterates between creating new representations (*engagement*) and then evaluating and choosing between them (*reflection*) – see Figure 6(a). By iterating between these two, creative output can emerge: the engagement stage can involve a degree of randomness, to encourage novel representations even though many will be of poor quality; the reflection stage can then choose the better ones. In this specific approach, the general E-R cycle is implemented in a distributed, stochastic manner, via a chaotic interaction of processes in a shared workspace, called a *bubble chamber*: this acts as the memory of the program, and contains the currently active representations (concepts, frames, and their instantiations) and allows them to be created, changed and evaluated by a series of independently acting software agents. These agents, called *codelets*, are stored on and selected from a data structure called a *coderack*, an idea taken from early computational creativity work (Hofstadter & FARG, 1995): each codelet can make a small change, e.g., suggesting a new structure that can be built given the current state, building or modifying a structure, evaluating its quality, or selecting one alternative vs. another. Codelets are chosen from the coderack with a degree of randomness determined by the program's *satisfaction*, a score of the quality of active structures in the bubble chamber. These measures of quality and overall satisfaction drive the overall emergent behaviour, with the best, most useful structures bubbling to the top of the program's attention as their activation increases.

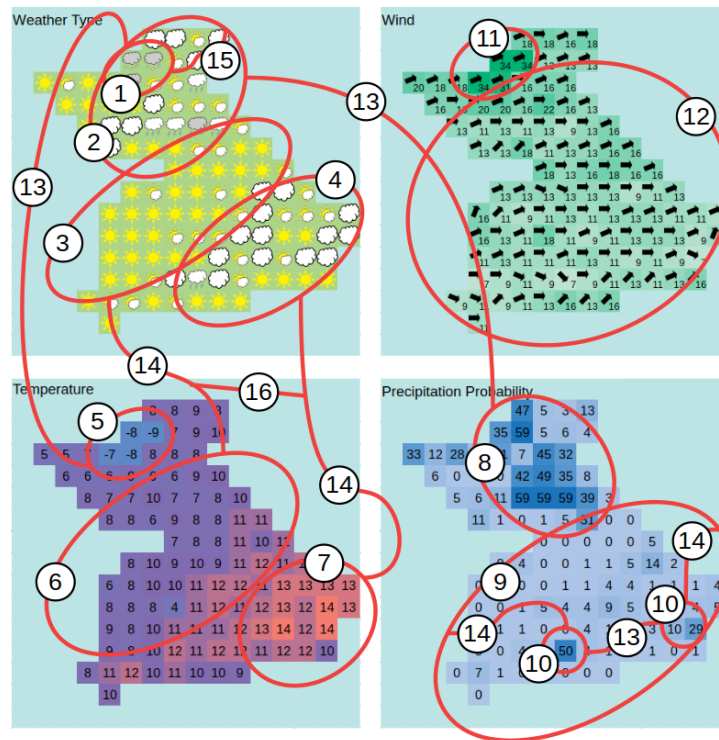


**Figure 6:** The conceptual architecture as a cycle of engagement and reflection.

The program's ability to monitor and evaluate the quality of the candidate representations it produces is therefore crucial, and for this it uses the concept of *views*, consistent ways of connecting one representation with another. The workspace contains not only *simplex views*, through which it can generate a text that describes a (piece of) input by connecting a candidate semantic representation with a frame that can be realised as text; but also *monitoring views*, through which it evaluates texts by parsing and checking the quality of the description against the original data (see Figure 6(b)). Given a set of basic conceptual frames (for example, the concepts of TEMPERATURE and LOCATION, and relations such as MORE, LESS and SAME), candidate texts can be built and constantly self-evaluated and compared in order to generate novel descriptions of an input dataset. Figure 7 shows an example of the conceptual frames and relations that can be inferred using this approach, given an input dataset in terms of quantitative

aspects of weather against geographical coordinates, with a possible high-quality output being a text such as the following:

It will be cloudy in the north with a high chance of rain and furthermore snow in the very north. There will be dry weather in the rest of the country but there may be pockets of rain in the south. It will be sunny in western and central areas but temperatures will be mild while it will be cloudy but warm in the southeast.



**Figure 7:** A four-channel map of weather in Britain with groups and relations. 1-12: Regions of similar weather; 13: AND relations; 14: BUT relations; 15: A FURTHERMORE relation; 16: A second-order AND relation. (Data from the Met Office).

An implementation of this architecture for the weather report domain is being implemented, and is currently able to generate simple descriptions of geographical variations in temperature. An evaluation of this with human judges showed that respondents ranked the computer-generated texts below human-generated texts for fluency, correctness, and completeness, but found the computer-generated texts easier to understand.

*This work is described in full in Wright & Purver (2020), attached here as Appendix D, Wright & Purver (2021b), attached here as Appendix E, and Wright & Purver (2021a), attached here as Appendix F.*

## 4 Headline Generation Using Sequence to Sequence Models

The approaches for headline generation described in Section 2 relied on modification of existing headlines, by injecting creative language and humorous replacements. In contrast, here we investigate whether it is possible to generate headlines using state-of-the-art neural encoder-decoder architectures. The employment of neural networks for natural language generation (NLG) has gained a lot of attention in recent years and several approaches were proposed. The main idea is to employ approaches that

have shown good performance in machine translation (or some other sequence to sequence task) and treat the generation task as a ‘translation’ between an input text (or sometimes even structured data) and the output text (Wen et al., 2015; Cho et al., 2014).

Most recent approaches towards headline generation consider it a summarization task and employ state-of-the-art neural summarization models for the task at hand. These models have been used to tackle several distinct variants of the headline generation task, such as bilingual headline generation (Shen et al., 2018), headlines for community question answering (Higurashi et al., 2018), multiple headline generation (Iwama & Kano, 2019) and also user-specific headline generation used in the recommendation systems (Liu et al., 2018). The summarization approach can also be upgraded to obtain headlines with a specific style. For example, Xu et al. (2019) first employed a Pointer Generator (Pointer-Gen) (See et al., 2017), a frequently used summarization model that takes a news article as an input and generates a headline. After that, they proposed a sensationalism scorer trained on a large dataset containing headlines with many comments (i.e. clickbaits) and employed auto-tuned reinforcement learning to reinforce the generation model to generate sensational clickbait headlines<sup>1</sup>. Jin et al. (2020) also proposed an approach to enrich the headlines obtained by a summarization system with three style options, humor, romance and clickbait, by combining the summarization and reconstruction tasks into a multitasking framework.

Encoder-decoder summarization approaches for language generation can under favorable circumstances produce texts of high quality. They are easier to adapt to a specific task at hand than rule-based approaches, i.e. by training the model on appropriate data. On the down side, these approaches are difficult to employ in real life scenarios. First, for successful training that leads to high-quality outputs, these systems require large amounts of data, which makes them infeasible in some low resource domains and languages (Gkatzia, 2016). Another problem related to the lack of data is a special type of overfitting called ‘hallucination’, where the system produces non-factual outputs that are not based on the data presented in the input (Reiter, 2018; Dušek et al., 2019). This severely limits the application of these systems in a news domain, where the production of factual text is essential. These systems also lack interpretability and their evaluation could be unreliable when not conducted manually by humans. Namely, it has been shown that commonly used automated evaluation metrics do not necessarily correlate well with human judges (Reiter & Belz, 2009; Dušek et al., 2018).

In the scope of the EMBEDDIA project, we acknowledged these shortcomings of neural NLG approaches and mainly focused on the development of rule-based and hybrid approaches, which can be used with less limitations in a real-life news media environment. Nevertheless, we do recognize the long term potential of these systems and therefore in this section we investigate whether it is possible to alleviate some of the debilitating deficiencies of neural approaches. More specifically, we use an encoder-decoder summarization system for automatic headline generation from the news article, as mentioned in the related work. However, the focus of the study is on the investigation of techniques for improving the performance of these systems in low-resource scenarios, which has, at least to our knowledge, not been inspected in many previous studies. Another distinction from most of the related work is that the proposed system is tested in a multilingual scenario, covering English and two EMBEDDIA languages, Croatian and Estonian.

## 4.1 Methodology

As mentioned above, we tackle the headline generation task as a sequence to sequence generation task, in which a neural model is trained to take a news article as an input and return the title of that specific article as an output. As a baseline, we test two state-of-the-art systems for summarization on the headline generation task. The first system is BART (Lewis et al., 2020), a denoising autoencoder for pretraining sequence-to-sequence models. BART employs a standard transformer-based neural

<sup>1</sup>Note that in the scope of EMBEDDIA project no aim to generate clickbaity headlines is being pursued due to questionable ethics of such products (Hindman, 2017). We are only presenting the references to such research as examples of successful style adaptation.



machine translation architecture and is pretrained on several noising tasks, in which the original text is corrupted and the model is trained to generate the uncorrupted output. To be more specific, the training corpus is corrupted by randomly shuffling the order of the original sentences or by using an in-filling scheme, where spans of text are replaced with a single mask token. BART achieved new state-of-the-art results on a set of tasks, ranging from classification, abstractive dialogue, question answering, and also summarization. The other approach we test was proposed in Lewis et al. (2020) and relies on the usage of the combination of pretrained transformer base language models, in our case two BERT models (Devlin et al., 2019). Using one language model as an encoder and the other as a decoder, the authors demonstrate the efficacy of pretrained language models for sequence generation, leading to state-of-the-art results on several tasks, including machine translation and text summarization. We name this approach BERT encoder-decoder (BERT-ED).

The distinct difference between the two approaches is that BART has already been pretrained as an encoder-decoder model on a large corpus consisting of books and Wikipedia (i.e. the same corpus as BERT). The BART model we used in our experiments was further fine-tuned for summarization on the CNN/Daily news summarization dataset (Hermann et al., 2015)<sup>2</sup>. The second proposed approach contains two pretrained BERT models<sup>3</sup> connected by a cross-attention model, which is not pretrained on any task in advance, but rather just randomly initialized. We suspect this difference would result in a different performance of both models when trained on a relatively small corpora. We hypothesise that while BART will be harder to fine-tune for a specific headline generation task due to its extensive pretraining as an encoder-decoder, it would nevertheless return semantically and grammatically better headlines. These headlines would most likely resemble summaries to a certain extent. Since the cross-attention layer in the approach composed of two BERT models has not been pretrained, this approach might require more training data to generate semantically and grammatically correct headlines. It would nevertheless be easier to adapt to a specific task and domain at hand.

As mentioned above, the main focus of the proposed research is to test whether these systems work in a low-resource setting. Most related work trained neural models on large datasets consisting of more than 100,000 documents. In contrast, we test the models in a low-resource setting, on datasets ranging from 10,000 to roughly 30,000 documents and investigate whether combining different pretraining and data augmentation techniques can improve the performance of the model. More specifically, we test three distinct pretraining techniques:

- **Text infilling:** As proposed by Lewis et al. (2020), about 20% of the training corpus is corrupted by an in-filling scheme, where spans of text are replaced with a single mask token. The encoder-decoder is then trained to generate the original text by being fed the corrupt input.
- **Sentence shuffling:** Same as in Lewis et al. (2020), sentences in the training corpus are randomly shuffled and the model is trained to generate the original text with the correct sentence order.
- **DBpedia first sentence prediction:** We use DBpedia (<https://www.dbpedia.org/>) to obtain Wikipedia descriptions for each Wikipedia entry that appears in the train set. The corpus of obtained Wikipedia descriptions is then used for model pretraining, in which the model is trained to generate first sentence of the description from the rest of the description. This task resembles a summarization task, since the first sentence in the Wikipedia descriptions tends to offer a summarized description of the subject, answering the two most basic questions for the nonspecialist reader: what (or who) is the subject and why is this subject notable (see Wikipedia guidelines for details). This pretraining task is only used for English experiments, since DBpedia does not cover the two EMBEDDIA languages used in our experiments, Croatian and Estonian.

Note that these pretraining techniques are only applied on the training data and no additional data is used. In this way, we inspect if the model's performance can be improved by proposing new training tasks instead of obtaining more data.

<sup>2</sup>More specifically, we used the 'facebook/bart-large-cnn' from the Huggingface library (<https://huggingface.co/>)

<sup>3</sup>For English we used two 'bert-base-uncased' models, for Estonian and Croatian we used the FinEst BERT and CroSloEngual BERT described in Ulčar & Robnik-Šikonja (2020), respectively.



Another way to obtain more data from the existing data is data augmentation. We employ five distinct techniques:

- **Word2vec augmentation:** For each news article in the train set, we replace random words in the articles by synonyms proposed by the Word2vec model trained on the Google News dataset. We employ the implementation in the TextAugment library<sup>4</sup> and we set the number of runs parameter to 5 and the probability of replacement to 0.3 (i.e. the algorithm will go five times through the text and try to augment each sentence with the 0.3 probability).
- **Wordnet augmentation:** Same as for Word2vec-based augmentation, we employ the implementation of the algorithm in the TextAugment library using the same parameters. The difference is that the replacement candidates are obtained from Wordnet.
- **EDA augmentation:** Here we apply the data augmentation approach proposed by Wei & Zou (2019). EDA consists of four operations: synonym replacement, random insertion, random swap, and random deletion. Again, we use the implementation of the algorithm in the TextAugment library.
- **Mixed augmentation:** We combine the approaches above in order to obtain the augmented text. Each article in the train set is first augmented with Word2vec. The augmented article is fed as an input to the EDA-based augmentation and the output of this augmentation is fed to the Wordnet-based augmentation.
- **BERT augmentation:** We test using BERT for data augmentation. 20% of words in the news article are masked and the masked article is fed to the BERT model, which proposes most probable candidate words for masked tokens.

For each original article in the train set, we generate 5 augmented articles using the algorithms described above. These new articles are inserted into the original training set and used for training of the headline generation model. We opted to generate five augmented texts per article because initial experiments suggested that using a smaller number results in an insufficient increase of the training dataset and using a larger number results in repetitions of the training examples.

For evaluation, we employ the ROGUE score, which is the current standard for evaluating generated summaries and headlines. However, it was shown in the past that ROGUE score does not necessarily have sufficient correlation with human judges (Reiter & Belz, 2009; Dušek et al., 2018) because it only compares n-gram overlap and is therefore agnostic of semantic similarity between true and generated headlines. To alleviate this problem, we propose two new evaluation measures, that also consider semantic similarity. The first measure, named semantic similarity (SS), measures cosine distance (CD) between the embedding of the true and generated headline. We employ sentence transformers (Reimers & Gurevych, 2019) for generating embeddings for true and generated headlines.<sup>5</sup>

Another evaluation approach is motivated by Wenpeng Yin & Roth (2019), who used a pretrained natural language inference (NLI) sequence-pair classifier as a zero-shot text classifier. We consider the true headline as the 'premise' and each generated headline as the 'hypothesis' and use the NLI model to predict whether the premise entails the hypothesis. We take the probability of the entailment between a true and a generated headline as a measurement of headline quality. Note that this measure is only used for English experiments, since there is no Croatian or Estonian model pretrained for NLI.<sup>6</sup>

<sup>4</sup><https://github.com/dsfsi/textaugment>

<sup>5</sup>More specifically, we employ the 'sentence-transformers/paraphrase-MiniLM-L6-v2' for experiments on English and 'sentence-transformers/paraphrase-xlm-r-multilingual-v1' for experiments on Croatian and Estonian. Both models are available in the Huggingface library.

<sup>6</sup>For English, we employ the 'typeform/distilbert-base-uncased-mnli' for entailment predictions from the Huggingface library.

## 4.2 Experiments

We conducted the experiments on three distinct datasets, namely the English KPTimes dataset (Gallina et al., 2019), and two datasets from EMBEDDIA media partners, the Estonian ExM news article dataset (Purver, Pollak, et al., 2021) and the Croatian 24sata news article dataset (Purver, Shekhar, et al., 2021). For English model training, we use both KPTimes train set, containing about 260,000 news articles, and the KPTimes validation set, containing 10,000 articles. For Croatian and Estonian, we use the same train and test splits as in our study about keyword extraction (Koloski et al., 2021). The dataset statistics are presented in Table 3.

For English, both BART and BERT-ED approaches are first tested in a high resource scenario, i.e. trained on the entire KPTimes train set. The results of these experiments are used for comparison of the two models and to obtain a reference point of how well these models work in an ideal scenario, to which we can compare results of our low-resource experiments. Next, both of these models are trained on the KPTimes validation set (i.e. we use the original KPTimes validation set as a train set in this scenario) without any additional pretraining or data augmentation. This reference point is used as a baseline to compare different pretrainings and data augmentations. The results of the experiments are presented at the top of Table 3. Note that in terms of all evaluation criteria, there is a large difference in performance between BERT-ED trained on the KPTimes validation set and BERT-ED trained on the KPTimes train set. Interestingly, there is only a rather marginal difference between BARTs trained on the KPTimes train and validation sets. While BERT-ED trained on the KPTimes train set outperformed BART by a large margin according to all ROGUE scores and offered comparable performance in terms of SS, BART outperforms BERT-ED by a comfortable margin according to NLI. We hypothesise that this means that BART does manage to generate relevant content, semantically close to the original title. The problem is in the form of the generated output, which more closely resembles a summary than a title. For example, the outputs are given in a sentence form and tend to be longer than original headlines.

Further manual investigation revealed that the initial hypothesis, which stated that BART is somewhat hard to fine-tune for a specific task of headline generation, was correct. All manually inspected produced headlines in both BART training scenarios very much resembled summaries. While the inspection suggested that BART manages to produce semantically and grammatically better headlines than BERT-ED (this is indicated by a slightly higher SS score), we still decided to exclude BART from further experiments due to following reasons: (1) the initial experiments suggested that the amount of fine-tuning data, and the employment of pretraining or data augmentation techniques had little to no effect on the model and the output it produced; (2) there is no BART model pretrained on Croatian or Estonian text, meaning that the model can not be employed for these two languages.

**Table 2:** News datasets used for empirical evaluation of headline generation.

| Language              | train set number of documents | test set number of documents |
|-----------------------|-------------------------------|------------------------------|
| English KPTimes train | 259,923                       | 10,000                       |
| English KPTimes valid | 10,000                        | 10,000                       |
| Croatian              | 32,223                        | 3,582                        |
| Estonian              | 10,750                        | 7,747                        |

The results of pretraining and data augmentation scenarios are presented in Table 3. When it comes to English data augmentation, all but one (Word2Vec augmentation) method manage to improve on the BERT-ED KPTimes valid baseline score. The biggest improvement can be observed for the BERT augmentation. Decent improvements according to all criteria can also be observed when EDA and Wordnet augmentation are used. For Croatian and Estonian, we only conduct BERT augmentation, since EDA and Wordnet augmentations are not available for these two languages and there is no pre-trained Word2Vec model trained on news. The improvements on both languages are consistent with the improvement gains obtained for English, when data augmentation results are compared to the baseline

**Table 3:** Results of the conducted multilingual experiments. Best results in the low-resource setting (i.e. excluding the BART and BERT-ED models trained on KPTimes train set) per language and per evaluation measure are in bold.

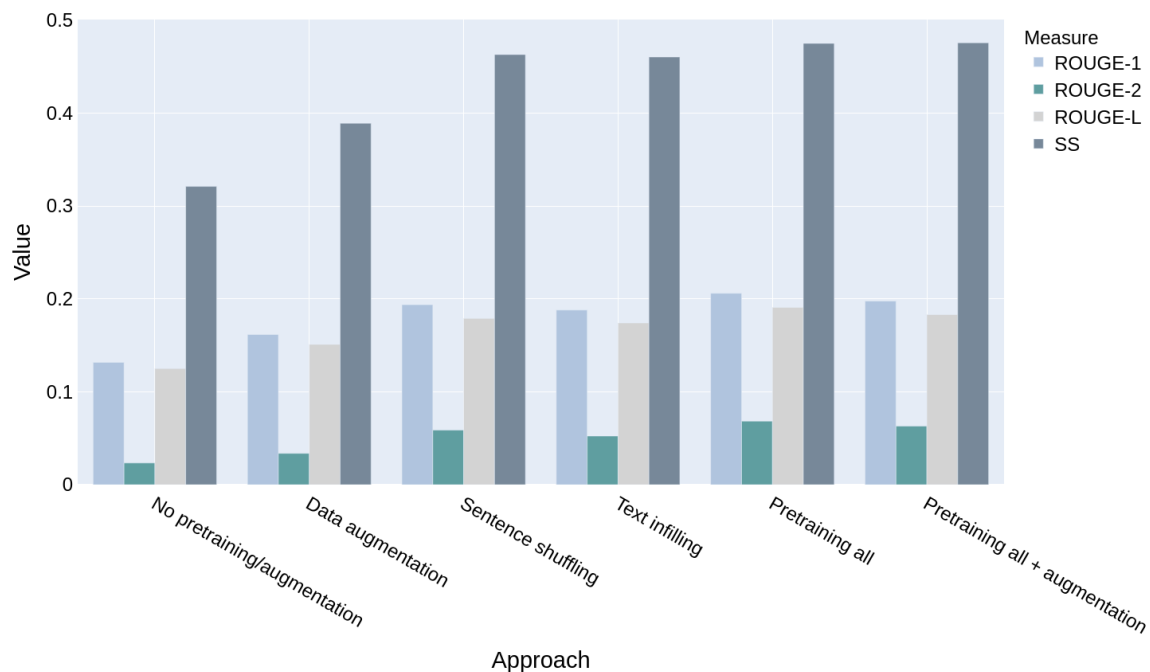
| Approach   | ROUGE-1      | ROUGE-2      | ROUGE-L      | SS           | NLI          |
|--|--------------|--------------|--------------|--------------|--------------|
| Baselines  |              |              |              |              |              |
| BART KPTimes train   | 0.225        | 0.073        | 0.202        | 0.554        | 0.433        |
| BART KPTimes valid   | 0.208        | 0.061        | 0.186        | 0.537        | 0.394        |
| BERT-ED KPTimes train  | 0.297        | 0.122        | 0.276        | 0.536        | 0.331        |
| BERT-ED KPTimes valid  | 0.135        | 0.026        | 0.129        | 0.269        | 0.131        |
| BERT-ED Croatian train   | 0.117        | 0.014        | 0.108        | 0.344        | /            |
| BERT-ED Estonian train   | 0.143        | 0.030        | 0.138        | 0.350        | /            |
| Data augmentation  |              |              |              |              |              |
| BERT-ED KPTimes valid BERT augmentation                          | 0.186        | 0.038        | 0.171        | 0.387        | 0.190        |
| BERT-ED KPTimes valid EDA augmentation                           | 0.171        | 0.037        | 0.159        | 0.358        | 0.173        |
| BERT-ED KPTimes valid MIX augmentation                           | 0.140        | 0.029        | 0.131        | 0.300        | 0.150        |
| BERT-ED KPTimes valid Word2Vec augmentation                      | 0.134        | 0.024        | 0.127        | 0.229        | 0.122        |
| BERT-ED KPTimes valid Wordnet augmentation                       | 0.174        | 0.037        | 0.162        | 0.369        | 0.173        |
| BERT-ED Croatian train BERT augmentation                         | 0.130        | 0.016        | 0.120        | 0.350        | /            |
| BERT-ED Estonian train BERT augmentation                         | 0.169        | 0.047        | 0.162        | 0.430        | /            |
| Pretraining  |              |              |              |              |              |
| BERT-ED KPTimes valid DBpedia pretraining                        | 0.153        | 0.031        | 0.144        | 0.324        | 0.152        |
| BERT-ED KPTimes valid sentence shuffling pretraining             | 0.224        | 0.065        | 0.207        | 0.444        | 0.238        |
| BERT-ED KPTimes valid text infilling pretraining                 | 0.198        | 0.048        | 0.183        | 0.414        | 0.210        |
| BERT-ED Croatian train sentence shuffling pretraining            | 0.158        | 0.045        | 0.144        | 0.432        | /            |
| BERT-ED Croatian train text infilling pretraining                | 0.159        | 0.038        | 0.144        | <b>0.454</b> | /            |
| BERT-ED Estonian train sentence shuffling pretraining            | 0.199        | 0.066        | 0.186        | 0.513        | /            |
| BERT-ED Estonian train text infilling pretraining                | 0.207        | 0.071        | 0.195        | 0.513        | /            |
| Combinations   |              |              |              |              |              |
| BERT-ED KPTimes valid all tasks pretraining                      | <b>0.239</b> | <b>0.076</b> | <b>0.219</b> | <b>0.468</b> | <b>0.262</b> |
| BERT-ED KPTimes valid all tasks pretraining + BERT augmentation  | 0.232        | 0.073        | 0.214        | 0.471        | 0.261        |
| BERT-ED Croatian train all tasks pretraining                     | <b>0.167</b> | <b>0.051</b> | <b>0.154</b> | 0.432        | /            |
| BERT-ED Estonian train all tasks pretraining                     | <b>0.212</b> | <b>0.078</b> | <b>0.199</b> | <b>0.525</b> | /            |
| BERT-ED Croatian train all tasks pretraining + BERT augmentation | 0.161        | 0.047        | 0.147        | 0.435        | /            |
| BERT-ED Estonian train all tasks pretraining + BERT augmentation | 0.200        | 0.069        | 0.188        | 0.521        | /            |

BERT-ED Croatian train and BERT-ED Estonian train approaches. The improvement is nevertheless the smallest for Croatian. This is most likely connected to the fact that the Croatian train set is about three times bigger, and therefore by default more appropriate for training of sequence-to-sequence models, than Estonian train and KPTimes valid datasets.

By pretraining the model on different tasks, we obtain substantial performance boosts. The sentence shuffling pretraining, in which we train the model to restore correct sentence order, offers the biggest improvement for English. For Croatian and Estonian, text infilling pretraining works slightly better. On the other hand, the DBpedia pretraining, which was only employed for English, offers a negligible improvement according to all criteria.

We also explore whether combining different tactics can further improve the results. By combining all three pretraining tasks for English (see ‘BERT-ED KPTimes valid all tasks pretraining’) or combining the two pretraining tasks for Estonian (see ‘BERT-ED Estonian train all tasks pretraining’), we manage to improve the results even further. Combining the two tasks for Croatian (see ‘BERT-ED Croatian train all tasks pretraining’) on the other hand does not improve the performance in terms of SS score but does slightly improve all three ROUGE scores. Interestingly, training the model pretrained on all tasks on the augmented datasets (see ‘BERT-ED KPTimes valid all tasks pretraining + BERT augmentation’, ‘BERT-ED Croatian train all tasks pretraining + BERT augmentation’ and ‘BERT-ED Estonian train all tasks pretraining + BERT augmentation’) instead of on the original dataset does not lead to further advancements.

The average performance for a specific approach across a set of three languages is visualized in Figure 8. It is visible that the employment of any of the pretraining or data augmentation techniques (or



**Figure 8:** Average performances of different pretraining and data augmentation approaches across three languages using four evaluation measures.

the combination of these techniques) leads to on average much better performance than conducting none (column No pretraining/augmentations). The pretraining nevertheless offers larger gains and one can also note that combining pretraining with data augmentation does generally not improve performance.

While improvements that can be achieved with pretraining or data augmentation are substantial, the best performance for English according to all measures is still achieved by the model trained on a large KPTimes train set. This indicates that currently there is still no sufficient substitution for a large dataset and also that there is still plenty of opportunity to improve the low-resource approach.

Examples of English headlines generated by the best approach in the low-resource setting, ‘BERT-ED KPTimes valid all tasks pretraining’, are presented Table 4. Note how some of the generated headlines contain non-factual information, i.e. hallucinations.

**Table 4:** Examples of English headlines generated with the best ‘BERT-ED KPTimes valid all tasks pretraining’ approach.

| True headline   | Generated headline  |
|---|---|
| iraq : ex - hussein aide convicted of terrorizing shiite kurds        | iraq : former foreign minister sentenced to 10 years      |
| vatican' s celestial eye, seeking not angels but data                 | a review of the vatican observatory                       |
| myanmar fighting edges toward china                                   | myanmar : shelling rebels in the border                   |
| as child migrants flood to border, u. s. presses latin america to act | u. s. moves to arrest child migrants from central america |
| when victory is impossible : reconciliation                           | u. s. and pakistan seek a deal with taliban               |
| merck wins u. s. approval for a new diabetes drug                     | u. s. approves new diabetes medicine                      |
| senate votes to add sexual orientation to hate crime protections      | senate to expand hate crimes law                          |
| alonso wins german grand prix   | fernando alonso wins german grand prix in germany         |
| judge urges president to address prison strike                        | judge rules on detainee at guantanamo                     |
| outcome of eric garner case bares a staten island divide              | eric garner dies at 87                                    |

## 5 Conclusions and further work

In this deliverable, we presented three distinct tactics for generation of creative language. First, we employed methods from the field of computational creativity to inject creative expressions into headlines and make the headlines more humorous. Next, creative architecture was employed to generate entire articles from the raw data. Finally, we proposed an encoder-decoder approach for headline generation, which is adapted for headline generation in low-resource settings.

When it comes to injection of creative expressions into existing texts, we approached the creative headline generation as a separate component from the news generation system. In the future it is important to bring these two systems into a closer collaboration in such a way that all news generated by the system contain creative traits. This requires extension of the computational creativity from headlines to the main text of a news article. For example, the creative architecture for news article generation described in Section 3 could benefit from introduction of colorful metaphors into the text.

Another planned extension of the creative language generation approach will focus on multilinguality. While multilinguality has been taken into account in Alnajjar et al. (2019), as the system can generate creative headlines in Finnish and English, the rest of the approaches based on computational creativity methods have focused on English only. This means that additional work is required to make our approaches truly multilingual. Nonetheless, experiments with metaphor generation in Finnish have shown very promising results suggesting that the methods are usable for other languages as well (Hämäläinen & Alnajjar, 2019).

The current work on sequence to sequence generation of headlines in low-resource setting is still in early stages. The main focus of the future work will be on improving the quality of generated headlines, by introducing novel pretraining tasks and data augmentation techniques. In order to do that, we will expand our evaluation setting, by introducing novel measures and also the manual evaluation. Finally, we will consider using these techniques for the generation of headlines with specific style, for example, humorous news titles.

## 6 Associated outputs

The work described in this deliverable has resulted in the following resources:

| Description  | URL   | Availability   |
|--|---|----------------|
| A parsed version of The Finnish News Agency Archive corpus provided by STT as part of EMBEDDIA | <a href="http://urn.fi/urn:nbn:fi:lb-2020031201">http://urn.fi/urn:nbn:fi:lb-2020031201</a>                                 | Limited Access |
| A code-base for a genetic algorithms approach for generating creative natural text             | <a href="https://github.com/EMBEDDIA/evolutionary-algorithm-for-NLG">github.com/EMBEDDIA/evolutionary-algorithm-for-NLG</a> | Public (MIT)   |

Parts of this work are also described in detail in the following publications, which are attached to this deliverable as appendices:

| Citation   | Status    | Appendix   |
|--|-----------|------------|
| Alnajjar, Khalid, Leo Leppänen, and Hannu Toivonen. No time like the present: methods for generating colourful and factual multilingual news headlines. Proceedings of the 10th International Conference on Computational Creativity. Association for Computational Creativity, 2019, pp. 258-265. | Published | Appendix A |
| Alnajjar, Khalid, and Hannu Toivonen. Computational generation of slogans. Natural Language Engineering, 27(5), 2021, pp. 575-607.   | Published | Appendix B |
| Alnajjar, Khalid, and Mika Hämäläinen. When a Computer Cracks a Joke: Automated Generation of Humorous Headlines. Proceedings of the 12th International Conference on Computational Creativity, 2021, pp. 292-299.   | Published | Appendix C |
| Wright, George, and Matthew Purver. Creative Language Generation in a Society of Engagement and Reflection. Proceedings of the Eleventh International Conference on Computational Creativity, 2020, pp. 169-172.   | Published | Appendix D |
| Wright, George, and Matthew Purver. Parsing Text in a Workspace for Language Generation. Proceedings of the 2021 Society for Text & Discourse Annual Conference, 2021, EasyChair Preprint no. 6171.  | Published | Appendix E |
| Wright, George, and Matthew Purver. Evaluating Natural Language Descriptions Generated in a Workspace-Based Architecture. Proceedings of the 12th International Conference on Computational Creativity, 2021, pp. 87-91.   | Published | Appendix F |

# References

- Alnajjar, K., & Hämmäläinen, M. (2021). When a computer cracks a joke: Automated generation of humorous headlines. In *Proceedings of the 12th international conference on computational creativity (iccc 2021)* (p. 292-299). Coimbra, Portugal.
- Alnajjar, K., Hämmäläinen, M., Chen, H., & Toivonen, H. (2017, June). Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the 8th international conference on computational creativity (iccc 2017)* (pp. 25–32). Georgia, Atlanta, United States: Georgia Institute of Technology.
- Alnajjar, K., Leppänen, L., & Toivonen, H. (2019, June). No time like the present: Methods for generating colourful and factual multilingual news headlines. In K. Grace, M. Cook, D. Ventura, & M. Maher (Eds.), *Proceedings of the 10th international conference on computational creativity* (pp. 258–265). Portugal: Association for Computational Creativity.
- Alnajjar, K., & Toivonen, H. (2021). Computational generation of slogans. *Natural Language Engineering*, 27(5), 575–607. doi: 10.1017/S1351324920000236
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1724–1734).
- Colton, S., & Wiggins, G. A. (2012). Computational creativity: The final frontier? In *Proceedings of the 20th european conference on artificial intelligence* (pp. 21–26). Amsterdam, The Netherlands: IOS Press.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N19-1423> doi: 10.18653/v1/N19-1423
- Dušek, O., Howcroft, D. M., & Rieser, V. (2019). Semantic noise matters for neural natural language generation. In *Proceedings of the 12th international conference on natural language generation* (pp. 421–426).
- Dušek, O., Novikova, J., & Rieser, V. (2018, November). Findings of the E2E NLG challenge. In *Proceedings of the 11th international conference on natural language generation* (pp. 322–328). Tilburg University, The Netherlands: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W18-6539> doi: 10.18653/v1/W18-6539
- Gallina, Y., Boudin, F., & Daille, B. (2019, October–November). KPTime: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th international conference on natural language generation* (pp. 130–135). Tokyo, Japan: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/W19-8617> doi: 10.18653/v1/W19-8617



- Gatti, L., Özbal, G., Guerini, M., Stock, O., & Strapparava, C. (2015). Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24th international conference on artificial intelligence* (pp. 2452–2458). AAAI Press. Retrieved from <http://dl.acm.org/citation.cfm?id=2832581.2832591>
- Gkatzia, D. (2016). Content selection in data-to-text systems: A survey. *arXiv preprint 1610.08375*.
- Hämäläinen, M., & Alnajjar, K. (2019, October–November). Let's FACE it. Finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th international conference on natural language generation* (pp. 290–300). Tokyo, Japan: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/W19-8637> doi: 10.18653/v1/W19-8637
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 1693–1701.
- Higurashi, T., Kobayashi, H., Masuyama, T., & Murao, K. (2018, August). Extractive headline generation based on learning to rank for community question answering. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1742–1753). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/C18-1148>
- Hindman, M. (2017). Journalism ethics and digital audience data. *Remaking the news: Essays on the future of journalism scholarship in the digital age*, 177–194.
- Hofstadter, D., & FARG. (1995). *Fluid concepts and creative analogies*. Basic Books.
- Hossain, N., Krumm, J., & Gamon, M. (2019, June). “president vows to cut <taxes> hair”: Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 133–142). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/N19-1012>
- Hossain, N., Krumm, J., Vanderwende, L., Horvitz, E., & Kautz, H. (2017, September). Filling the blanks (hint: plural noun) for mad Libs humor. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 638–647). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D17-1067> doi: 10.18653/v1/D17-1067
- Iwama, K., & Kano, Y. (2019, October–November). Multiple news headlines generation using page metadata. In *Proceedings of the 12th international conference on natural language generation* (pp. 101–105). Tokyo, Japan: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W19-8612> doi: 10.18653/v1/W19-8612
- Jin, D., Jin, Z., Zhou, J. T., Orii, L., & Szolovits, P. (2020, July). Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5082–5093). Online: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2020.acl-main.456> doi: 10.18653/v1/2020.acl-main.456
- Koloski, B., Pollak, S., Škrlić, B., & Martinc, M. (2021). Keyword extraction datasets for Croatian, Estonian, Latvian and Russian 1.0. *Ekspress Meedia Group*.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., . . . Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (p. 7871–7880). Online: Association for Computational Linguistics.
- Liu, T., Li, H., Zhu, J., Zhang, J., & Zong, C. (2018). Review headline generation with user embedding. In *Chinese computational linguistics and natural language processing based on naturally annotated big data* (pp. 324–334). Springer.



- Lynch, G. (2015, 12). Every word you set: Simulating the cognitive process of linguistic creativity with the PUNdit system. *International Journal of Mind Brain and Cognition*, 6(1-1).
- Purver, M., Pollak, S., Freienthal, L., Kuulmets, H.-A., Krustok, I., & Shekhar, R. (2021). Ekspress news article archive (in estonian and russian) 1.0. *Ekspress Meedia Group*.
- Purver, M., Shekhar, R., Pranjić, M., Pollak, S., & Martinc, M. (2021). 24sata news article archive 1.0. *Styria Media Group*.
- Pérez y Pérez, R., de Cossío, M. G., & Guerrero, I. (2013). A computer model for the generation of visual compositions. In *Proceedings of the fourth international conference on computational creativity* (p. 105-112). Sydney, Australia.
- Pérez y Pérez, R., & Sharples, M. (2001). MEXICA: A computer model of a cognitive account of creative writing. *JETAI*, 13.
- Reimers, N., & Gurevych, I. (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing*. Association for Computational Linguistics. Retrieved from <https://arxiv.org/abs/1908.10084>
- Reiter, E. (2018). *Hallucination in Neural NLG*. <https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/>. (Accessed: 2020-03-02)
- Reiter, E., & Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4), 529–558.
- See, A., Liu, P. J., & Manning, C. D. (2017, July). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1073–1083). Vancouver, Canada: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/P17-1099> doi: 10.18653/v1/P17-1099
- Sharples, M. (1998). *How we write: Writing as creative design*. Routledge.
- Shen, S.-q., Chen, Y., Yang, C., Liu, Z.-y., Sun, M.-s., et al. (2018). Zero-shot cross-lingual neural headline generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12), 2319–2327.
- Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multi-lingual models. In P. Sojka, I. Kopeček, K. Pala, & A. Horák (Eds.), *Text, speech, and dialogue TSD 2020* (Vol. 12284). Springer. Retrieved from [https://doi.org/10.1007/978-3-030-58323-1\\_11](https://doi.org/10.1007/978-3-030-58323-1_11)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
- Veale, T., & Li, G. (2013a). Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 660–670). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from <http://aclweb.org/anthology/P13-1065>
- Veale, T., & Li, G. (2013b). Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 660–670). Sofia, Bulgaria: Association for Computational Linguistics.
- Wei, J., & Zou, K. (2019, November). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 6383–6389). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://www.aclweb.org/anthology/D19-1670>
- Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D., & Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1711–1721).

- Wenpeng Yin, J. H., & Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. Hong Kong, China. Retrieved from <https://arxiv.org/abs/1909.00161>
- Wright, G., & Purver, M. (2020, September). Creative language generation in a society of engagement and reflection. In F. A. Cardoso, P. Machado, T. Veale, & J. M. Cunha (Eds.), *Proceedings of the 11th International Conference on Computational Creativity (ICCC)* (p. 169-172). Online / Coimbra, Portugal: Association for Computational Creativity. Retrieved from [http://computationalcreativity.net/iccc20/papers/ICCC20\\_Proceedings.pdf](http://computationalcreativity.net/iccc20/papers/ICCC20_Proceedings.pdf)
- Wright, G., & Purver, M. (2021a, September). Evaluating natural language descriptions generated in a workspace-based architecture. In A. Gómez de Silva Garza, T. Veale, W. Aguilar, & R. Pérez y Pérez (Eds.), *Proceedings of the 12th International Conference on Computational Creativity (ICCC)* (p. 87-91). Online: Association for Computational Creativity. Retrieved from [https://computationalcreativity.net/iccc21/wp-content/uploads/2021/09/ICCC\\_2021\\_paper\\_97.pdf](https://computationalcreativity.net/iccc21/wp-content/uploads/2021/09/ICCC_2021_paper_97.pdf)
- Wright, G., & Purver, M. (2021b, August). Parsing text in a workspace for language generation. In *Proceedings of the 2021 Society for Text & Discourse Annual Conference*. EasyChair Preprint no. 6171. Retrieved from <https://easychair.org/publications/preprint/l4c9>
- Xiao, P., Alnajjar, K., Granroth-Wilding, M., Agres, K., & Toivonen, H. (2016). Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the 7th international conference on computational creativity (iccc 2016)*. Paris, France: Sony CSL.
- Xu, P., Wu, C.-S., Madotto, A., & Fung, P. (2019, November). Clickbait? sensational headline generation with auto-tuned reinforcement learning. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3065–3075). Hong Kong, China: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/D19-1303> doi: 10.18653/v1/D19-1303

# Appendix A: No time like the present: methods for generating colourful and factual multilingual news headlines

## No Time Like the Present: Methods for Generating Colourful and Factual Multilingual News Headlines

**Khalid Alnajjar**

khalid.alnajjar@helsinki.fi

**Leo Leppänen**

leo.leppanen@helsinki.fi

**Hannu Toivonen**

hannu.toivonen@helsinki.fi

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

### Abstract

News headlines are the main method for briefly providing a summary of the news article and attracting an audience. In this paper, we experiment with different existing methods for producing colourful expressions and news headlines computationally, in a practical setting. Our case study is conducted by modifying an automated journalism system that generates multilingual news in three languages, namely English, Finnish and Swedish. We adapt existing methods for creative headlines and figurative language generation into the headline generation process of the system, modifying them to work in a multilingual setting. We conduct our evaluation by asking online judges to assess the original titles produced by the unmodified system and those enhanced by the methods described in this paper. The results of the evaluation suggest that the presented methods increase the creativity of existing headlines while maintaining their descriptiveness.

### Introduction

The interest in automated journalism has increased in the past years, driven by the ability to produce tailored stories cost-effectively even for small audiences, i.e., the so-called long tail effect. Current methods for automated news generation typically utilize linguistic templates written by journalists, and fill them in with appropriate information from structured data sources. Template-based methods give strong control over the output generated by the system and ensures conveying the message as intended. However, news produced by these approaches tend to be repetitive and sound mechanistic.

Headlines are an essential part of the news. They must relate to the news article and briefly describe it while motivating readers to visit and read the article. Automated journalism systems aim to produce informative headlines, but not colourful ones requiring creativity.

In this paper, we experiment with different existing methods for creating colourful expressions and with their use in template-based news headlines. We seek for a balance between creativity and factuality. Because of the latter, we build on an existing template-based system that produces factual headlines; for the former, we generate creative expressions and add them to the factual headlines.

For our case study, we use a modified version of *Valteri*<sup>1</sup> (Leppänen et al. 2017), an automated journalism system, as the baseline. *Valteri* generates election news about the 2017 Finnish municipal election results in three languages, English, Finnish and Swedish. For the scope of this work, we focus on two languages only: English and Finnish.

Creativity, such as use of figurative language, is something human journalists consider to be one of their strengths when compared to automated journalism systems (van Dalen 2012). Creativity is missing from most, if not all, automated journalism systems is creativity. This also applies to *Valteri*.

Inspired by previous research on generating figurative language (Veale and Li 2013; Alnajjar et al. 2017) and creative headlines (Lynch 2015; Gatti et al. 2015), we present two methods which add a creative touch to news headlines generated by the automated journalism system. The methods are developed to operate in a multilingual setting. The first method finds a suitable well-known phrase (e.g. movie title) to be presented to the reader as a catchy title (i.e. it draws attention) along with the factual message. The other method injects figurative phrases (e.g. similes and metaphors) into headlines, depending on the polarity of the news. We exploit recent research in word cross-lingual embeddings, permitting us to project knowledge from English, with rich linguistic resources, into a less-resourced one, i.e. Finnish.

In our evaluation, we crowdsourced the assessment of the headlines to online judges (acting as the audience). We asked them to evaluate the original headline produced by *Valteri* and the new altered headlines produced by the methods described in this paper in order to test the applicability of these methods in a practical scenario. The judges were asked to assess aspects such as informativeness, correctness and catchiness, to measure the effects of figurative modifications on the original headlines. Because of the availability of crowdsourcing workers, the current evaluation is conducted on English headlines only and Finnish is left for future work.

This paper is structured as follows. We begin by reviewing related work on headline generation. Thereafter, we describe the *Valteri* system and how the creative component is attached to the system. We then elucidate the methods employed by us to convert the headlines generated by *Valteri*

<sup>1</sup> <https://www.vaalibotti.fi/>

into more colourful ones. The evaluation details are then provided, followed by the results. Lastly, we discuss the results and conclude this work.

### Related Work

Previous research on headline generation is extensive, covering different approaches based e.g. on rules, statistics, summarization or machine learning. In this section, we briefly describe the most relevant work.

Hedge Trimmer (Dorr, Zajic, and Schwartz 2003), a rule-based method for headline generation, decides which words are to be retained and which to be pruned from the news article. Their rules are linguistically motivated and based on analyzing human-made headlines written in English. Building such rules is tedious, especially when dealing with multilingual news articles. Wang, Dunnion, and Carthy (2005) extended the work by introducing a C5.0 decision tree classifier for predicting which words to include in the title.

Zajic, Dorr, and Schwartz (2002) use a Hidden Markov Model to generate news headlines for a news story by having the model capture keywords from the beginning, i.e. first paragraphs, of the story. A Viterbi Decoding algorithm is then applied to headlines generated by the model to find the most representative headline. Additionally, four decoding parameters are imposed to ensure the quality of the generated headline, namely: (1) a length penalty, to keep headlines within the 5 to 15 word length limits, (2) a position penalty, to give a higher penalty to words appearing later in the story, (3) a string penalty, to encourage neighbouring words and (4) a gap penalty, to reduce the distance between selected words. Another statistical approach (Colmenares et al. 2015) uses sequence prediction methods for learning how humans craft headlines. Given a story, their model classifies whether a certain token in the story should be in the headline or not. In the case of a token being classified as in-headline, their method considers various features regarding the text of the story, the token (e.g. parts-of-speech tags and name-entities) and the constructed headline at each stage. Other statistical-based research on headline generation has been conducted by Banko, Mittal, and Witbrock (2000), Knight and Marcu (2002), Wan et al. (2003) and Unno et al. (2006).

Summarization-based techniques treat the problem of headline generation as producing a one sentence digest of the article (Morita et al. 2013; Martins and Smith 2009; Filippova 2010). Summarization techniques tend to extract and then compress sentences existing in the new article, which results in reusing words/phrases existing in the article. Furthermore, deep learning models have also been employed in the generation of headlines by learning how to summarize a certain text (Ayana et al. 2016). Such models require sufficiently big training data sets which can be prohibitively large for some scenarios.

A way of expressing headlines in various styles is to learn different ways of talking about the same news article. Wubben et al. (2009) have proposed a way of grouping news articles from different sources based on the content similarity. Using the different ways of writing a headline for a certain topic, a machine translation model could be trained to learn how to paraphrase headlines (Wubben, Bosch, and

Krahmer 2010). *HEADY* (Alfonseca, Pighin, and Garrido 2013), on the other hand, performs event pattern clustering and generates a headline for an unseen news article by inferring headlines based on the events in it.

The above approaches do not consider an important aspect of news headlines, which is catchiness. To our best knowledge, catchiness in news headlines generation is addressed only in the work by Lynch (2015) and Gatti et al. (2015).

Lynch (2015) proposed a system for adding a well-known phrase (e.g. songs, films ... etc) as a prefix to an existing title. The added phrase is intended to catch the attention of the readers and increase search engine optimization. The system extracts keywords from an article, clusters and expands them. Then, it pairs keywords from distinct clusters if they co-occurred in a corpus of 5-grams. Using a pseudo-phonetic string matching algorithm and semantic similarity measurement, the system finds and ranks well-known phrases suitable for the pair. Lastly, it embeds the matched well-known phrase in the existing headline.

In the method described by Gatti et al. (2015), titles are given a creative touch by blending them with well-known expressions. Their headline generation process extracts keywords from the input news article. Thereafter, the method finds existing well-known phrases that are semantically similar to the existing headline and the article. These phrases are then modified by altering a word in them that satisfies a semantic similarity threshold, and lexical and syntactic constraints.

Despite the advances in automated headline generation, research on generating catchy and diverse headlines for automated journalism is scarce, especially in a multilingual setting with less-resourced languages.

### Adding Creativity to Valtteri Headlines

*Valtteri* (Leppänen et al. 2017; Melin et al. 2018) is a multilingual system for automated journalism, reporting on the 2017 Finnish municipal elections. The system follows a data-driven approach to generate news while ensuring certain requirements, e.g. accuracy (i.e. factual and not misleading) of the produced news.

We add the creativity component to the system at a central stage of the pipeline, immediately after the aggregation process. It has access to the data and the selected templates to be used in the news. The component can alter the content of the news article produced along with its headline.

Inspired by existing research on computational linguistic creativity and creative headline generation, we implement two methods for producing colourful headlines. The methods are:

1. **Phrase-copying:** We find and insert a suitable well-known phrase into a factual headline (Lynch 2015; Gatti et al. 2015).
2. **Figurative-injection:** We generate figurative expressions using linguistic patterns and knowledge-bases of stereotypical properties of nouns (Veale and Li 2013; Alnajjar et al. 2017), and insert them into existing headlines.

For our use case, these methods should be incorporated in the automated journalism system and they should work

in multiple languages. To achieve this, in case the required linguistic resources are not available for Finnish, we resort to pre-trained and aligned multilingual word embeddings  $\zeta$  (Bojanowski et al. 2017; Joulin et al. 2018). In these models, a vector representation of a word in a certain language (e.g. *king* in English,  $\zeta_{en}$ ) should roughly point to the same semantic direction in another model (e.g. *kuningas* in Finnish,  $\zeta_{fi}$ ) and vice versa. With the help of these aligned models, we can exploit available linguistic creativity resources in English and project them into Finnish.

The following sub-sections describe the two methods for colourful headline generation in-depth.

### Phrase-copying: Insertion of Well-Known Phrases

Inspired by the research by Lynch (2015) and Gatti et al. (2015), we implement a method for finding and inserting well-known phrases into headlines produced by *Valtteri*. The results have the form “*phrase: headline*”, c.f. Table 1 for examples. Juxtapositioning the phrase with the headline is expected to catch the attention of viewers and motivate them to click on the headline to read the news article, while keeping the factual content of the original headline intact. For this to work as intended, the method should find a well-known phrase that matches the original headline.

We use two types of well-known phrases: proverbs and movie titles. Proverbs for each language are extracted from [wikiquote.org](https://en.wikiquote.org/)<sup>2</sup>. Regarding movie titles, we use the dataset of movies provided by IMDB<sup>3</sup>. We restrict the dataset to movies with more than 100,000 votes, to exclude generally unfamiliar titles. As these titles are in English and we desire to know how they are known to people in other languages, we query *Wikipedia* with the movie title in English and retrieve the title of its corresponding Finnish *Wikipedia* article. As an example, the movie title “Harry Potter and the Philosopher’s Stone” is known to Finns as “Harry Potter ja viisasten kivi”.

We perform a preprocessing step on the collected phrases to clean and expand them. The process commences by stripping punctuation and any parentheses including the content in them to omit some explanations given in the proverbs. We also removed phrases containing more than 5 words to avoid lengthy headlines that could distract the audience. Some movie titles separate a general title and a subtitle by a colon or a dash; we include in our dataset both the short version (before the colon/dash) and the long version (all of the text).

In total, the database of well-known phrases contains 1,744 and 1,322 phrases in English and Finnish, respectively. We denote this database by  $P$ .

In order to identify a well-known phrase that matches the headline, two aspects are checked: 1) semantic similarity (or relatedness) between the phrase and the headline, and 2) prosody of the phrase and the headline. Semantic similarity is used for coherence of the resulting combination, while

prosody is evaluated to increase catchiness of the result.

We employ a greedy algorithm to match phrases to a given headline  $H$ . For each phrase  $\rho$  in  $P$ , the method computes the cosine semantic similarity between individual words  $w_1, w_2$  in  $\rho$  and  $H$ , using the corresponding language model  $\zeta_l$ , where  $l$  is either ‘en’ or ‘fi’, as follows:

$$sim_{words}(w_1, w_2, t, l) = \begin{cases} \zeta_l(w_1, w_2), & \text{if } \zeta_l(w_1, w_2) \geq t \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$sim_{phrases}(H, \rho, t, l) = \sum_{w \in H} \sum_{k \in \rho} sim_{words}(w, k, t, l) \quad (2)$$

In equations 1 and 2,  $t$  is a threshold for the minimum semantic similarity desired. We empirically set  $t$  to 0.3. While increasing the threshold  $t$  would find phrases that are more semantically similar to the headline, it would also narrow the space of possible solutions, especially for languages other than English. If a phrase has received a semantic similarity score on Equation 2 greater than 0, it is considered to be similar to the headline.

When a phrase  $\rho$  is found to be semantically similar to headline  $H$ , the method computes four prosody features between the matched phrase and the headline. These features are assonance, consonance, alliteration and rhyme. We utilize the *espeak-ng tool*<sup>4</sup> to acquire the International Phonetic Alphabet (IPA) transcriptions of words. The tool supports producing IPA for multiple languages, including English and Finnish. For each word pair in  $\rho \times H$ , the method evaluates whether the pair has phonetic similarity of any of the four prosody features. Then, each prosody feature is aggregated to phrase-level by computing its average score over the word pairs in  $\rho \times H$ .

Finally, we aggregate all four features into one number by obtaining a weighted sum of their phrase-level scores. We assigned to rhyme and alliteration weights of 40% each, and to consonance and assonance weights of 10% each, based on empirical testing.

Given a headline  $H$ , the method considers phrases  $\rho$  in  $P$  in a random order, computing the above measures of semantic similarity and prosody. The method keeps progressing until it finds ten well-known phrases that have a positive semantic similarity and a positive phonetical similarity with headline  $H$ . Among the ten phrases, the method then picks the phrase with the highest prosody score. The magnitude of the semantic similarity is not considered, since the template-based headline generation method tends to give the highest semantic similarity to the same phrases; for prosody, there is more variation based on how the template has been instantiated.

Finally, the selected phrase is inserted into the headline. For headline examples generated by this method, see Table 1.

<sup>2</sup>English: [https://en.wikiquote.org/wiki/English\\_proverbs\\_\(alphabetically\\_by\\_proverb\)](https://en.wikiquote.org/wiki/English_proverbs_(alphabetically_by_proverb))  
Finnish: [https://en.wikiquote.org/wiki/Finnish\\_proverbs](https://en.wikiquote.org/wiki/Finnish_proverbs)

<sup>3</sup><https://datasets.imdbws.com/>

<sup>4</sup><https://github.com/espeak-ng/espeak-ng>



| #   | Baseline  | Phrase-copying   | Figurative-injection   |
|-----|---|--|--|
| (1) | Most seats go to The Centre Party of Finland in Kangasniemi       | Legends of the Fall: Most seats go to The Centre Party of Finland in Kangasniemi   | Most seats go to The Centre Party of Finland, the free queen, in Kangasniemi                 |
| (2) | Biggest vote gains for The Green League in Kuopio                 | Alls well that ends well: Biggest vote gains for The Green League in Kuopio        | Biggest vote gains for The Green League –the lovely god– in Kuopio                           |
| (3) | Biggest gains for The Christian Democrats across Lapin vaalipiiri | The Running Man: Biggest gains for The Christian Democrats across Lapin vaalipiiri | Biggest gains for The Christian Democrats, as powerful as a soldier, across Lapin vaalipiiri |
| (4) | Second largest gains for The Christian Democrats in Rovaniemi     | The Transporter: Second largest gains for The Christian Democrats in Rovaniemi     | Second largest gains for The Christian Democrats –the king– in Rovaniemi                     |
| (5) | The Finns Party lose three seats in Jyväskylä                     | To each his own: The Finns Party lose three seats in Jyväskylä                     | Like a spy, The Finns Party lose three seats in Jyväskylä                                    |

Table 1: Five examples of generated headlines from an existing headline by the two presented methods in this paper, in English.

### Figurative-injection: Generation of Figurative Language

The figurative-injection method inserts figurative language (e.g. metaphors and similes) into existing headlines. See the column ‘Figurative-injection’ of Table 1 for examples of headlines generated by this method. We next describe the method.

If the given headline has polarity with respect to the main entity in the headline, a political party or candidate in our case, then the method adds a figurative comparison to an adjective and common noun that is stereotypically associated with the polarity. The aim is that this comparison indirectly attributes properties to the entity of the headline, thereby emphasizing the polarity in a creative, figurative way.

Given that the automated journalism system works with structured data and given templates, we can directly associate polarities with the templates and values used to populate them, and avoid the need for automated polarity analysis of headlines. The polarity is determined by inspecting the reported result (i.e. the gains or losses of votes and seats) in the headline, while taking negations into account. In the cases where the headline states that an entity has received a positive result (e.g. majority of votes, biggest gains ... etc) or negative result (e.g. no seats, lose X seats ... etc) it is classified accordingly; otherwise, it is considered to be neutral. Neutral headlines are not modified by this method.

Identification of suitable adjectives and common nouns proceeds in three steps, performed once as a pre-processing step. First, we have manually listed seed nouns that match the election domain (e.g. win, success; loss, defeat). Second, we use corpus-based methods to identify adjectives associated to the seed nouns (e.g. heroic; tragic). Third, we identify common nouns that are stereotypically associated to these adjectives, using an existing knowledge base. We next detail these steps.

First, we manually define a set of seed nouns describing each of the polarities:

- *positive*: win, gain, accomplishment, success, achievement
- *negative*: loss, defeat, failure

Second, using the seed words, we mine stereotypical properties related to them. We observe trigrams in Google N-Grams (Brants and Franz 2006) that match the linguistic pattern “a/n \* SEED”, where SEED is any of the seed words, as conducted in previous research by Veale and Li (2013). We retrieve the adjectival properties that occur at the wildcard position (“\*”) in such trigrams. We then use the resource by Alnajjar et al. (2017) to prune out noisy and non-adjectival relations (e.g. “a 3-5 win”). Examples of mined properties for the two categories are: “a *heroic* achievement” and “a *tragic* loss”.

Some positive adjectives can be associated with negative situations (e.g. “a *great* loss”). We use the polarity function provided in *Pattern* library (De Smedt and Daelemans 2012) to predict the polarity of adjectives, and we filter out any adjectival property that has a polarity which does not match the intended classification.

Third, the method looks for suitable metaphorical nouns (common nouns in our case) that are strongly associated with the desired properties. For this, we use a tested dataset  $\kappa$  of nouns and their weighted stereotypical properties (Alnajjar et al. 2017). An example of a noun and its stereotypical properties along with their weights is *King*: {powerful: 1563, successful: 1361, ... etc}.

Given a headline to modify, the method now has access to knowledge of which properties describe a positive or negative situation and which nouns are well-known to possess these properties. The method then searches for a suitable metaphorical noun to be introduced in the headline. It does so by iterating over all the properties describing the situation and the common nouns in  $\kappa$  to find out which nouns are associated to many of these properties. In the process, the method keeps track of all these nouns and how strongly they are related to the relevant properties in knowledge-

base  $\kappa$ . Thereafter, the nouns are sorted based on the sum of their association weights. A random noun having a total weight above the third quartile of weights is selected to be the metaphorical noun. A random stereotypical property of the selected noun is then chosen while ensuring that it meets two constraints: 1) it is strongly associated with the noun (i.e. in the top 50%) and 2) it describes the situation. The selected noun and its property will be used, in the remainder of this method, to construct a figurative expression.

The knowledge-base  $\kappa$  and the linguistic pattern used to find adjectival properties are in English but we desire to generate figurative language in multiple languages. To overcome this obstacle, we employ aligned word embedding models between multiple languages (English and Finnish) as follows. When the method is requested to generate a figurative expression for a language other than English, it begins by using the trigrams and knowledge available in English to find suitable a suitable noun and property. Once a noun and a property are selected, the method obtains their vector representations in the English model. These vectors are then projected into the other aligned model (i.e. Finnish). We consider the closest word to the projected vector as the representation of the word in the other language.

To realize a figurative expression using the selected metaphorical noun and property, we hand-crafted a set of figurative templates in both languages, given in Table 2. For each template, we define whether the template should be injected in the headline before or after the name of the entity. Depending on the position of the entity's name in the headline, a random figurative template is chosen.

| English                                      | Finnish                              | Position |
|--|--------------------------------------|----------|
| , as <i>PROPERTY</i> as [a\ n] <i>NOUN</i> , | , <i>PROPERTY</i> kuin <i>NOUN</i> , | after    |
| , the <i>NOUN</i> ,                          | , <i>NOUN</i> ,                      | after    |
| –the <i>NOUN</i> –                           | – <i>NOUN</i> –                      | after    |
| , the <i>PROPERTY NOUN</i> ,                 | , <i>PROPERTY NOUN</i> ,             | after    |
| –the <i>PROPERTY NOUN</i> –                  | – <i>PROPERTY NOUN</i> –             | after    |
| Like [a\ n] <i>NOUN</i> ,                    | Kuin <i>NOUN</i> konsanaan,          | before   |
| Like [a\ n] <i>PROPERTY NOUN</i> ,           | Kuin <i>PROPERTY NOUN</i> ,          | before   |

Table 2: Hand-crafted figurative templates in English and Finnish to be injected in existing headlines. The position column indicates whether the template should be injected before or after the entity name.

Finally, the chosen template gets filled with the selected noun and property. To ensure producing grammatically correct metaphorical expressions, we use *Pattern* to reference nouns and properties correctly, for English. Regarding Finnish, we analyze and inflect the projected words in the Finnish space into the nominative form, if necessary, using *UralicNLP* (Hämäläinen 2019) and *Omorfi* (Pirinen 2015).

## Evaluation

We asked online judges on [figure-eight.com](http://figure-eight.com) to evaluate both the baseline (non-creative) headlines produced by *Valtteri* and the modified (creative) headlines by the methods

described above. As Finnish is not supported by the crowdsourcing platform, we only evaluated English headlines at this stage.

Our evaluation dataset is constructed as follows. We randomly selected a pair of a location and an entity in Finland and passed them to *Valtteri* to obtain the news article covering the election results of the entity in that location, in English. For locations, we only considered the ones on country, district or municipality levels, to exclude news for small areas. In case the reported news by *Valtteri* was classified to be neutral in its polarity, then another random pair was selected. This process was repeated until we had 100 news articles.

The headline of each generated news article was then passed to the creativity component, which generated two modified headlines using the two presented methods. Table 1 shows examples of headlines generated by the methods.

Overall, the evaluation dataset contains 300 English headlines: 100 from the baseline system and 100 generated by both methods. We asked 10 online judges to evaluate each headline. Judges were given a brief description of the task, and the first paragraph of the news story generated by *Valtteri*. They were then asked to evaluate the headline on a 5-point Likert scale against the following claims:

1. The headline is descriptive of the article.
2. The headline is grammatically correct.
3. The headline is catchy.
4. The headline is creative.
5. The headline can be considered offensive.
6. The headline is generated by a computer.

Some of these perspectives are from the prior research by Lynch (2015) and they should be self-explaining.

The quality control mechanism enforced in crowdsourcing was that a minimum of 10 seconds was spent in answering questions about five headlines, in order to eliminate spammers that answer them randomly. We did not apply other measures since the questions and interpretations are subjective and do not have correct answers.

## Results

The evaluation process resulted in 3,000 unique judgments from crowdsourcing, 1,000 for each type of headlines. Table 3 shows the mean and standard deviation of judgments received on each question for the three types of headlines, and Figure 1 gives the diverging bar charts for the answers. We next look at the results for each property assessed.

**Descriptive** From the results, it appears that the three types of generated headlines were considered to be descriptive on average (i.e.  $\mu_x > 3$ ). Despite all versions of the headline having the same factual message present, the headlines produced by *Valtteri* (the baseline) were judged to be the most descriptive. This difference is statistically significant.

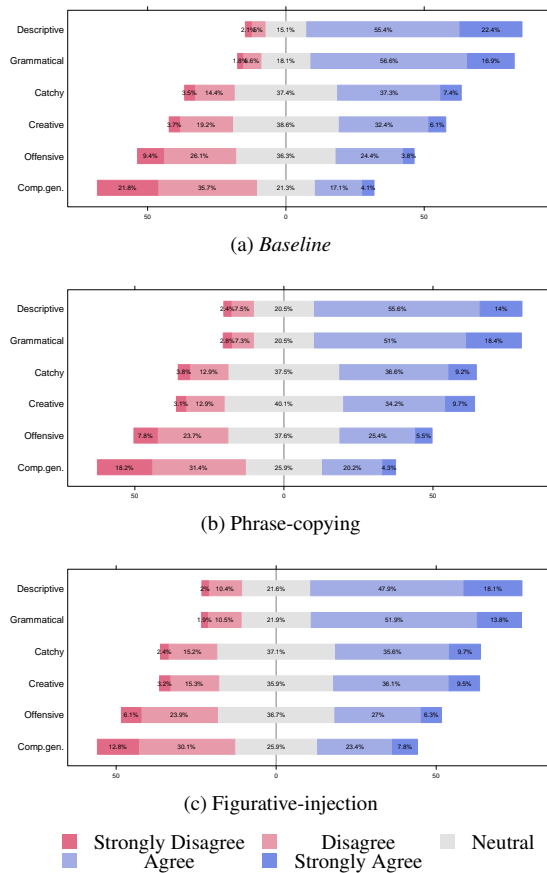


Figure 1: Diverging bar charts illustrating the percentage of judgments received on each question for the three types of headlines.

**Grammatical** The results concerning the grammaticality of produced headlines is similar to the results on their descriptiveness. That is, all methods produced grammatically correct headlines on average, with headlines produced by the baseline being statistically significantly the most grammatically correct.

**Catchy** In terms of the catchiness of the headlines, both non-baseline methods have slightly improved the catchiness of original headlines. The difference, however, is not statistically significant.

**Creative** Regarding the creativity of headlines, the phrase-copying method is perceived to be the most creative, on average. The judges deemed both non-baseline methods to be more creative than the baseline (with statistical significance), as both methods have increased the agreements by approximately 6%.

**Offensive** Headlines produced by the non-baseline methods are more likely to produce offensive headlines. The difference between the baseline and the proposed methods is statistically significant. However, headlines are generally neutral and not offensive (i.e.  $\mu_x \leq 3$ ).

**Generated** Headlines produced by the non-baseline methods are considered to be computer-generated more often than the ones generated by the baseline methods, to a statistically significant degree. However, headlines produced by all variations could pass as being written by humans as the majority of judges believed that they are not generated by computers.

## Discussion

The aim of the proposed methods was to add creative language to news headlines, in order to add variation to them and to make them more interesting for readers.

According to our empirical results, the proposed methods indeed improved the creativity of the original headlines produced by *Valteri*. This shows that the methods had some success in making the headlines more creative.

By adding creative elements, we also aimed to make the headlines more catchy. Here the methods were only slightly successful: catchiness was improved marginally. This result shows that creativity does not necessarily improve catchiness in the case of headline generation.

The modified headlines lost some of the descriptiveness of the original headlines, indicating that the added elements did not match the contents of the headline or the news story. In the case of the phrase-copying method, the main problem seems to be that despite our aim to choose phrases that are semantically related to the original headline, the added phrases can still be poorly chosen. Our measure of semantic similarity considers relations between individual words, but does not in any way take into account the meanings or mental images of the phrases as a whole. Adding a phrase with polarity matching the polarity of the headline could help, but more work is needed to make better use of well-known phrases given their rich, cultural meanings and interpretations. For the figurative-injection method, the result suggests that the selection of nouns and adjectives, but also the design of the templates used to inject figurative expressions, should be improved.

The modified headlines also lost some of their grammatical correctness. This is somewhat surprising for the phrase-copying method whose results consist of a well-known phrase and the original headline. Technically speaking, one would expect these to be grammatically about equally correct with the original headlines. A possible explanation is that the decrease in perceived grammatical correctness is influenced by poor matching of the added phrase and the original headline, as discussed above. An alternative cause is that the judges did not recognize all “well-known” phrases and therefore did not see the (grammatical) point in the generated headline. In the case of the figurative-injection method, the result implies again that the templates used to inject figurative expressions should be improved for grammatical fluency.



|                      | Descriptive |      | Grammatical |      | Catchy      |      | Creative     |      | Offensive   |      | Comp.gen.   |      |
|----------------------|-------------|------|-------------|------|-------------|------|--------------|------|-------------|------|-------------|------|
|                      | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ | $\mu_x$      | $SD$ | $\mu_x$     | $SD$ | $\mu_x$     | $SD$ |
| <i>Baseline</i>      | <b>3.91</b> | 0.87 | <b>3.80</b> | 0.86 | 3.31        | 0.93 | 3.18         | 0.94 | <b>2.46</b> | 1.13 | <b>2.87</b> | 1.01 |
| Phrase-copying       | 3.75*       | 0.93 | 3.71*       | 0.88 | <b>3.35</b> | 0.95 | <b>3.35*</b> | 0.93 | 2.61*       | 1.12 | 2.97*       | 1.01 |
| Figurative-injection | 3.70*       | 0.95 | 3.65*       | 0.91 | <b>3.35</b> | 0.93 | 3.33*        | 0.95 | 2.83*       | 1.15 | 3.04*       | 1.00 |

Table 3: The mean  $\mu_x$  and standard deviation  $SD$  of judgments received for each type of generated headlines on the six questions. The best result for each question appears in boldface.

\* The value is statistically significantly different ( $p < 0.05$ ) from the value for the baseline headline (non-parametric permutation test with one hundred million repetitions, one-tailed, not corrected for multiple testing).

We also assessed whether the modified headlines are more likely to be offensive than the original headlines. This turned indeed to be the case. By inspecting the headlines which were considered to be the most offensive, we noticed that they were usually negative expressions generated by the figurative-injection method. By construction, the method compares a party or person to a common noun, and therefore negative analogs easily become offensive to the involved party. The two most offensive headlines are 1) “No seats for The Christian Democrats, the thief, in Nousiainen” and 2) “Like a fool, The Finns Party drop most seats in Mynämäki”. This result and examples highlight that care needs to be taken when using automated creativity methods to talk about persons (or parties), in order to avoid unintentional offensive expressions. The proposed methods could be modified to reduce the chances of producing offensive outputs as follows: 1) introduce a dictionary of taboo words to filter out risky words or well-known phrases containing them and 2) use a lower threshold when searching for metaphorical nouns, in order to allow for a wider selection of (safe) words. A better but bigger change would be to produce figurative comparisons to the events in the news, such as loss of seats, rather than to the persons or parties involved. Nevertheless, the final output cannot be guaranteed to be safe for production unless it is verified by a human.

Finally, the modified headlines generated by the proposed methods were recognized to be computer-generated more often than the original (computer-generated) headlines. This suggests that the methods to select and inject materials need to be improved, as the eventual goal is produce headlines that appear less computer-generated than the baseline method.

### Conclusion

In this paper, we have presented methods for modifying an existing headline generation method, in order to give the headlines a creative touch. The methods work by inserting well-known phrases or figurative language in the headline templates. In our use case, we extended the headline generation method of *Valteri*, a system that generates news reports on the 2017 Finnish elections in English, Finnish and Swedish. We also described how the methods can utilize cross-lingual links between Wikipedia articles and aligned multilingual word embedding models in order to take advan-

tage of English resources when producing Finnish headlines, but this aspect was not evaluated due to lack of crowdsourcing workers.

Our empirical evaluation using English headlines generated by the proposed methods shows that they made the headlines more creative, and also slightly more catchy, but at the same time we observed a decrease in how descriptive and grammatically correct the headlines are.

In future work, we plan to improve the methods to select and inject materials to headlines, taking better into account the implied meanings of the added phrases or expressions, as well as making the results linguistically more fluent. Evaluation of Finnish headlines will help assess how well the cross-lingual aspects of the methods work. Interesting topics for future work also include automatic extraction of templates for injection of figurative expressions, and production of apt, yet ethically appropriate, figurative expressions. Finally, it would be interesting to introduce figurative language in the body of automatically generated news, not only headlines.

### Acknowledgments

This work has been supported by European Union’s Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media).

### References

- Alfonseca, E.; Pighin, D.; and Garrido, G. 2013. Heady: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1243–1253. Sofia, Bulgaria: Association for Computational Linguistics.
- Alnajjar, K.; Härmäläinen, M.; Chen, H.; and Toivonen, H. 2017. Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the 8th International Conference on Computational Creativity*, 25–32. Atlanta, United States: Georgia Institute of Technology.
- Ayana; Shen, S.; Liu, Z.; and Sun, M. 2016. Neural headline generation with minimum risk training. *CoRR* abs/1604.01904.

- Banko, M.; Mittal, V. O.; and Witbrock, M. J. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, 318–325. Hong Kong: Association for Computational Linguistics.
- Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.
- Brants, T., and Franz, A. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA. Philadelphia, PA.
- Colmenares, C. A.; Litvak, M.; Mantrach, A.; and Silvestri, F. 2015. Heads: Headline generation as sequence prediction using an abstract feature-rich space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 133–142. Denver, Colorado: Association for Computational Linguistics.
- De Smedt, T., and Daelemans, W. 2012. Pattern for Python. *Journal of Machine Learning Research* 13:2063–2067.
- Dorr, B.; Zajic, D.; and Schwartz, R. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*.
- Filippova, K. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 322–330. Beijing, China: Coling 2010 Organizing Committee.
- Gatti, L.; Özbal, G.; Guerini, M.; Stock, O.; and Strapparava, C. 2015. Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, 2452–2458. AAAI Press.
- Hämäläinen, M. 2019. Uralicnlp: An NLP library for Uralic languages. *Journal of Open Source Software* 4(37):1345.
- Joulin, A.; Bojanowski, P.; Mikolov, T.; Jégou, H.; and Grave, E. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2979–2984. Brussels, Belgium: Association for Computational Linguistics.
- Knight, K., and Marcu, D. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.
- Leppänen, L.; Munezero, M.; Granroth-Wilding, M.; and Toivonen, H. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, 188–197. Santiago de Compostela, Spain: Association for Computational Linguistics.
- Lynch, G. 2015. Every word you set: Simulating the cognitive process of linguistic creativity with the PUNdit system. *International Journal of Mind Brain and Cognition* 6(1-1).
- Martins, A. F. T., and Smith, N. A. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, ILP '09, 1–9. Boulder, Colorado: Association for Computational Linguistics.
- Melin, M.; Bäck, A.; Södergård, C.; Munezero, M. D.; Leppänen, L. J.; and Toivonen, H. 2018. No landslide for the human journalist—an empirical study of computer-generated election news in finland. *IEEE Access* 6:43356–43367.
- Morita, H.; Sasano, R.; Takamura, H.; and Okumura, M. 2013. Subtree extractive summarization via submodular maximization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1023–1032. Sofia, Bulgaria: Association for Computational Linguistics.
- Pirinen, T. A. 2015. Development and use of computational morphology of Finnish in the open source and open science era: Notes on experiences with omorf development. *SKY Journal of Linguistics* 28:381–393.
- Unno, Y.; Ninomiya, T.; Miyao, Y.; and Tsujii, J. 2006. Trimming CFG parse trees for sentence compression using machine learning approaches. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, 850–857. Sydney, Australia: Association for Computational Linguistics.
- van Dalen, A. 2012. The algorithms behind the headlines. *Journalism Practice* 6(5-6):648–658.
- 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*, 660–670. Sofia, Bulgaria: Association for Computational Linguistics.
- Wan, S.; Dras, M.; Paris, C.; and Dale, R. 2003. Using thematic information in statistical headline generation. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, 11–20. Sapporo, Japan: Association for Computational Linguistics.
- Wang, R.; Dunnion, J.; and Carthy, J. 2005. Machine learning approach to augmenting news headline generation. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*.
- Wubben, S.; van den Bosch, A.; Krahmer, E.; and Marsi, E. 2009. Clustering and matching headlines for automatic paraphrase acquisition. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, 122–125. Athens, Greece: Association for Computational Linguistics.
- Wubben, S.; Bosch, A. v. d.; and Krahmer, E. 2010. Paraphrasing headlines by machine translation: Sentential paraphrase acquisition and generation using google news. *LOT Occasional Series* 16:169–183.
- Zajic, D.; Dorr, B.; and Schwartz, R. 2002. Automatic headline generation for newspaper stories. In *Workshop on Automatic Summarization*, 78–85. Philadelphia, PA, USA: Association for Computational Linguistics.



## Appendix B: Computational generation of slogans

*Natural Language Engineering* (2020), 1–33  
doi:10.1017/S1351324920000236

CAMBRIDGE  
UNIVERSITY PRESS

### ARTICLE

## Computational generation of slogans

Khalid Alnajjar\*  and Hannu Toivonen 

Department of Computer Science and HIIT, University of Helsinki, Helsinki 00014, Finland

\*Corresponding author. E-mail: [khalid.alnajjar@helsinki.fi](mailto:khalid.alnajjar@helsinki.fi)

(Received 13 May 2019; revised 2 April 2020; accepted 3 April 2020)

### Abstract

In advertising, slogans are used to enhance the recall of the advertised product by consumers and to distinguish it from others in the market. Creating effective slogans is a resource-consuming task for humans. In this paper, we describe a novel method for automatically generating slogans, given a target concept (e.g., car) and an adjectival property to express (e.g., elegant) as input. Additionally, a key component in our approach is a novel method for generating nominal metaphors, using a metaphor interpretation model, to allow generating metaphorical slogans. The method for generating slogans extracts skeletons from existing slogans. It then fills a skeleton in with suitable words by utilizing multiple linguistic resources (such as a repository of grammatical relations, and semantic and language models) and genetic algorithms to optimize multiple objectives such as semantic relatedness, language correctness, and usage of rhetorical devices. We evaluate the metaphor and slogan generation methods by running crowdsourced surveys. On a five-point Likert scale, we ask online judges to evaluate whether the generated metaphors, along with three other metaphors generated using different methods, highlight the intended property. The slogan generation method is evaluated by asking crowdsourced judges to rate generated slogans from five perspectives: (1) how well is the slogan related to the topic, (2) how correct is the language of the slogan, (3) how metaphoric is the slogan, (4) how catchy, attractive, and memorable is it, and (5) how good is the slogan overall. Similarly, we evaluate existing expert-made slogans. Based on the evaluations, we analyze the method and provide insights regarding existing slogans. The empirical results indicate that our metaphor generation method is capable of producing apt metaphors. Regarding the slogan generator, the results suggest that the method has successfully produced at least one effective slogan for every evaluated input.

**Keywords:** Natural language generation; Slogan generation; Metaphor generation; Computational creativity

### 1. Introduction

Slogans are memorable short phrases that express an idea. They are frequently used in advertising and branding to enhance the recall of products by customers and to distinguish it from competitors. For example, the phrase “Connecting People” triggers many of us to think of *Nokia*. This effect of associating a phrase (i.e., slogan) with a concept (i.e., brand) highlights the significance of slogans in advertising. Coming up with successful slogans is challenging, both for humans and machines. This paper proposes a method to draft advertising slogans computationally.

Advertising professionals often resort to rhetorical devices to create memorable and catchy slogans. A common such rhetorical device is metaphor; *Nokia*’s “Connecting People” and *Red Bull*’s “Red Bull gives you wings” are both examples of metaphoric slogans. The subtle metaphor in *Nokia*’s slogan paints an image of mobile devices establishing intimate relations between people, in addition to providing a concrete means of communication between them. *Red Bull*’s slogan is more obviously metaphoric since a drink cannot give wings. An interpretation of the metaphor is

that the drink helps you exceed your physical limits. According to Reinsch (1971), metaphors and similes make messages more persuasive.

We propose a novel method for generation of metaphorical candidate slogans for a given target concept (e.g., *car*) and property (e.g., *elegant*). The intended use of the method is to assist advertising professionals during brainstorming sessions, not to substitute the professionals. Examples of slogans created by the method for the above input include “The Cars Of Vintage,” “Travel Free,” “The Cars Of Stage,” and “Travel, Transport and Trip.” Behind each such generated slogan is a computationally generated metaphor, intended to associate the property *elegant* to cars. For instance, the slogan “The Cars Of Stage” establishes an analog between cars and dancers, suggesting that cars are as elegant as dancers.

Our work contributes to the fields of Natural Language Processing (NLP) and Natural Language Generation (NLG) in two ways. On the one hand, this work computationally processes and generates short expressions. On the other hand, the focus is on figurative language, especially generation and evaluation of some figurative devices.

More specifically, the contributions are the following: (1) We start by providing a characterization and review of the field of slogan generation, also including some other creative expressions. (2) Our main contribution is a novel computational method for generation of slogans. (3) A key component of the slogan generation method is an algorithm for generating metaphors, a novel contribution in itself. (4) We evaluate the proposed method, including the metaphor generator, and provide extensive experimental results based on crowdsourcing. A partial description of a preliminary version of the method is given by Alnajjar, Hadaytullah, and Toivonen (2018).

This paper is structured as follows. We begin by covering the necessary conceptual background regarding slogans and metaphor in Section 2, and we review related work on slogan and metaphor generation in Section 3. In Section 4, we describe a novel method for generating metaphorical slogans. We report on empirical evaluation in Section 5 and discuss the results in Section 6. Section 7 contains our conclusions.

## 2. Background

*Slogans.* We define a slogan, from an advertising perspective, as a concise, advertisable, and autonomous phrase that expresses a concept (e.g., an idea, product, or entity); the phrase will be frequently repeated and associated with the concept. Elements of advertisability include creativity, catchiness (i.e., draws attention), memorability (i.e., easy to memorize and recall), clearness (i.e., does not cause confusion), informativeness (i.e., has a message), and distinctiveness (i.e., uniqueness) (Dahl 2011). Creating slogans that exhibit these elements manifests the difficulty of the task.

Slogans change over time and typically are not fixed for advertising campaigns (Kohli, Suri, and Thakor 2002). Brands may change their slogans, for instance, to target a certain audience, provide a new persuasive selling statement for a given product, or reflect changes in the company’s values. Mathur and Mathur (1995) have found that firms that change their slogan seem to have positive effects on their market value. Continuous change of slogans can benefit from a slogan generator such as the one introduced in this paper.

Slogans, taglines, and mottoes are similar to the extent that they are considered synonyms. Slogans and taglines are often used interchangeably; however, slogans are made for an advertising campaign whereas taglines are employed as an identifiable phrase for the brand. In other words, a slogan is made for one or more advertising campaigns but a tagline is typically made once for the lifetime of the company. On the other hand, mottoes are sayings that represent a group’s (e.g., corporate, political, and religious) vision such as *Google*’s previous motto “Don’t be evil”.<sup>a</sup> In this

<sup>a</sup> From <https://en.wikipedia.org/wiki/Google>.

paper, we use the term slogan to refer to all these collectively, given the similarities they have and the difficulties in distinguishing them.

*Rhetorical devices.* Language is a device for communication; slogans convey a message to the receiver, usually a persuasive one about a concept (i.e., product, service, or company). Like poems, slogans have a stylistic language concerned with *how* a message is expressed. Rhetorical devices such as figures of speech are examples of stylistic language. They exploit the listeners' knowledge of the language and persuade them by redirecting their thinking toward a path intended by the speaker.

Many slogans employ rhetorical devices. For instance, *Yellow Page's* slogan "Let your fingers do the walking" uses personification expressing fingers as entities capable of walking. Previous research suggests that slogans employing rhetorical devices tend to be favored and remembered better by consumers (Reece, Van den Bergh, and Li 1994). Moreover, different rhetorical devices in slogans have various effects on consumers. For instance, Burgers *et al.* (2015) suggest that slogans containing conventional metaphors are liked and considered more creative than slogans containing irony.

*Metaphor.* Metaphor is a figurative expression where some properties get implicitly highlighted or attributed from one concept to another one. For instance, the metaphor "time is money" implies that time is valuable without saying it directly: by equating *time* and *money*, the property *valuable* of *money* is attributed to the concept *time*. Other interpretations are also possible, as is usual with metaphors.

A metaphor involves two concepts, a tenor and a vehicle (Richards 1936). In "time is money," *time* is the tenor and *money* is the vehicle. As another example, in *Oakmont Bakery's* slogan "We create delicious memories,"<sup>b</sup> the tenor (pastry) is implicitly compared to memorable events (e.g., a wedding), implying that it is their cakes that make the event remembered for long. In this example, like in many slogans, the tenor and vehicle are not mentioned explicitly but rather must be inferred from the context. A nominal metaphor, on the other hand, is a metaphor in the simple form "tenor is [a\n] vehicle." "Time is money" is an example of a nominal metaphor.

Multiple theories exist in the literature about metaphors, providing us with guidance into what characteristics are exhibited by metaphors and what makes a metaphor apt. The *salience imbalance theory* (Ortony *et al.* 1985; Ortony 1993) states that metaphoricity occurs when the tenor and vehicle share attributes but some are highly salient for the vehicle only, and this imbalance causes these attributes to be highlighted by the metaphorical expression. Tourangeau and Sternberg (1981) argue that *similarities* within and between the domain of the vehicle and that of the tenor are aspects humans consider when comprehending metaphors. Katz (1989) points out that *concrete* vehicles that are *semantically moderately distant* from the tenor result in apt metaphors. An important property of metaphors is that they are *asymmetrical* in the sense that the metaphor "A is B" highlights different properties than "B is A."

*Analysis of slogans.* Reece, Van den Bergh, and Li (1994) have analyzed linguistic characteristics of slogans, in addition to other characteristics such as their themes, to find out how they affect receivers in recalling the brand. Their study indicates that utilizing linguistic devices has indeed affected the recall of the brand. The top eight slogans with high recall contained the following linguistic devices: (1) self-reference (i.e., having the brand name in the slogan), (2) alliteration, (3) parallel construction (i.e., repeating rhythm or words from the first phrase in the second phrase), (4) metaphor, and (5) use of a well-known phrase. The authors have also noticed that the slogan with the highest number of correct brand identifications made use of rhymes. As a result,

<sup>b</sup> Slogan examples in this paper are from <http://www.textart.ru/>, unless otherwise specified.



these linguistic devices seem to have a significant influence on recalling the brand, albeit, some of the frequently found linguistic devices in slogans did not have such outstanding influence, for example, puns.

Inspired by the analysis and taxonomy of linguistic devices used by Reece, Van den Bergh, and Li (1994), Miller and Toman (2016) manually analyzed slogans from various linguistic perspectives, focusing on rhetorical figures and covering other linguistic devices. Their research shows that linguistic devices existed in 92% of 239 slogans, out of which 80% and 42% were schematic and tropic rhetorical devices, respectively. Additionally, the two most common rhetorical devices which were found in figurative slogans are phonetic and semantic devices, covering 87% and 37% of them, respectively. Some phonetic devices appeared more than others, for example, both consonance and assonance occurred in 59% of figurative slogans whereas 32% and 4% of them had alliteration and rhyming, respectively. The semantic device with the highest frequency is metaphor, existing in 24% of rhetorical slogans. Other linguistic devices analyzed by the authors are syntactic, orthographic, and morphological devices which appeared in less than 30 slogans.

A similar manual analysis was conducted by Dubovičienė and Skorupa (2014). Their results also demonstrate that slogans use rhetorical devices frequently, especially figurative language and prosody. However, the percentages of individual rhetorical devices do not match the one by Miller and Toman (2016), which could be due to the difference in the analysis method and the sources of slogans used during the analysis.

Tom and Eves (1999) have found that advertisements containing rhetorical figures are more persuasive and have higher recall in comparison to slogans that do not utilize rhetorical figures. A research conducted by Reece, Van den Bergh, and Li (1994) suggests that recalling a slogan relies largely on the slogan itself, not on the advertising budget, years in use or themes. Furthermore, advertising slogans tend to contain positive words (Dowling and Kabanoff 1996) which would give the receiver a positive feeling about the brand.

*Problem definition.* We define the task of slogan generation from a computational perspective as follows. Given an input concept/tenor  $T$  (e.g., car) and an adjectival property  $P$  (e.g., elegant), produce slogans that associate concept  $T$  with property  $P$ . As a reminder from the beginning of this section, a slogan is a concise, advertisable, and autonomous phrase, where advertisable often implies creativity, catchiness, memorability, or related properties. “Car is elegant,” an obvious output for the example task, clearly is not a good slogan.

As the above background on slogans indicates, slogans tend to include rhetorical devices. Among the schematic and tropic rhetorical devices, prosody and metaphor were found to be the most frequent devices (Miller and Toman 2016). Motivated by this, as well as by their effectiveness in enhancing the recall of the brand (Reece, Van den Bergh, and Li 1994), we focus on these two types of rhetorical devices. Besides the usage of rhetorical devices, slogans have positive sentiment and, as a rule, should neither carry negative words nor communicate negative meanings.

The specific slogan generation task that we consider in this paper is the following. Given an input concept/tenor  $T$  and an adjectival property  $P$ , produce positive slogans that are related to  $T$ , that metaphorically associate concept  $T$  with property  $P$ , and/or that are prosodic. An interesting subtask in its own right is to find a metaphorical vehicle  $v$  that attributes property  $P$  to concept/tenor  $T$  when the concept/tenor is equated with the vehicle.

### 3. Related work

Research on computational generation of creative expressions is relatively scarce. In this section, we briefly review related work on generating nominal metaphors and on generation of slogans and other creative expressions.

### 3.1. Computational generation of metaphors

*Metaphor Magnet*,<sup>c</sup> a web service built by Veale and Li (2012), generates and interprets metaphors by observing the overlap of stereotypical properties between concepts. The metaphor generation process accepts a tenor as input. It uses knowledge regarding properties strongly associated with the tenor to find other concepts, potential vehicles, that share those properties. The aptness of the potential metaphors is measured in the process. The interpretation model, in turn, looks at strongly associated properties shared by the two input concepts (a tenor and a vehicle) and returns the salient features among them. Metaphor Magnet is based on a knowledge base of stereotypical associations, obtained using Google 3-grams and web searches with suitably designed linguistic patterns.

Galvan *et al.* (2016) generate metaphors based on categorizations of concepts and adjectival properties associated with them, as provided by the *Thesaurus Rex* web service (Veale and Li 2013). Their method takes the tenor as input, picks one of its properties at random, and then identifies a vehicle that highlights that property. The vehicle identification starts by finding a suitable category: one that is (strongly) associated to both the tenor and the property. A concept falling in the selected category and with a strong association to the selected property is then chosen as the vehicle.

Xiao and Blat (2013) propose a method for generating pictorial metaphors for advertisements. Their approach takes a concept and a list of adjectival properties to express, and uses multiple knowledge bases, for example, word associations and common-sense knowledge,<sup>d</sup> to find concepts with high imageability. The found concepts are then evaluated against four metrics, namely affect polarity, salience, secondary attributes, and similarity with the tenor. Concepts with high rank on these measures are considered apt vehicles to be used metaphorically.

In contrast to the direct metaphor generation methods above, we employ a metaphor interpretation model to identify apt metaphors that are more likely to result in the desired meaning. The interpretation model, Meta4meaning (Xiao, Alnajjar, Granroth-Wilding, Agres, and Toivonen 2016), uses corpus-based word associations to approximate properties of concepts. Interpretations are obtained by considering salience of the properties of the tenor and the vehicle, either their aggregation or difference.

### 3.2. Computational generation of slogans

Strapparava, Valitutti, and Stock (2007) propose a “creative function” for producing advertising messages automatically. The function takes a topic and a familiar expression as input, and modifies the expression by substituting some words with new ones related to the given topic. In the process, they use semantic and emotional relatedness along with assonance measures to identify candidate substitutes. This approach is motivated by the “optimal innovation hypothesis” (Giora 2003). The hypothesis states that 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.

Özbal, Pighin, and Strapparava (2013) introduce a framework called *BrainSup* for creative sentence generation. The framework generates sentences such as slogans by producing expressions with content semantically related to the target domain, emotion, and color, and some phonetic properties. Using syntactical treebanks 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, Tomašič *et al.* (2014) and Tomašič, Žnidaršič, and Papa (2015) propose an approach for generating slogans without any user-defined target words by extracting

<sup>c</sup> <http://ngrams.ucd.ie/metaphor-magnet-acl/>.

<sup>d</sup> ConceptNet: <http://www.conceptnet.io>.

keywords from the textual description of the target concept. Their evaluation criteria are different from *BrainSup*'s evaluation, and they use genetic algorithms instead of beam search.

The approach proposed by Žnidaršič, Tomašič and Papa (2015) employs case-based reasoning where actual slogans written by humans (not their syntactical skeletons) were reused with some modifications in a different context as a new slogan (cf. the approach of Strapparava, Valitutti, and Stock (2007) earlier in this section). The approach commences by retrieving slogans related to the textual description of the input concept using semantic similarities. Slogans are then transformed by replacing content words in them with words from the concept description while satisfying existing part-of-speech (POS) tags.

The *Bislon* method by Repar, Martinc, Znidarsic, and Pollak (2018) produces slogans based on cross-context associations, so-called bisociations (Koestler 1964), and prosody features (alliteration, assonance, consonance, and rhyme). The method accepts three types of input—a set of documents, *Metaphor Magnet* terms (Veale and Li 2012), or domain-specific terms—for both the main concept and the bisociated one. Keywords are automatically extracted from the input and then expanded using a word-embedding model. To generate slogans, the method uses existing slogans as skeletons and fills them with candidate words that match the POS tags of the placeholders. The method ranks slogan candidates based on their relevance to the input and their semantic cohesion as estimated by a language model. Finally, the top slogan candidates are suggested to the user.

In terms of slogan generation in languages other than English, Yamane and Hagiwara (2015) propose a method for producing Japanese taglines related to the input theme and keywords specified by the user. The method generates slogan candidates from a large-scale  $n$ -gram corpus containing words related to the input. The candidates are then assessed on three aspects: (1) the relatedness of words, (2) grammaticality (based on POS  $n$ -grams), and (3) novelty (based on combinations of words). The highest scoring candidates are output to the user. Another approach for producing Japanese slogans is proposed by Iwama and Kano (2018).

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

Persuasive messages are not only used in slogans, but news headlines also employ them a lot to encourage the audience to read the article (Fuertes-Olivera *et al.* 2001). Gatti *et al.* (2015) have demonstrated how well-known expressions (such as slogans) can be utilized to produce interesting news headlines. Their headline generation process extracts keywords from a news article and then alters man-made slogans based on semantic similarities, dependency statistics, and other criteria, resulting in catchy news headlines.

The method proposed in this paper differs from existing methods for slogan generation in a couple of important aspects. First, it focuses on a specific marketing message, that is, generating slogans for a product while expressing a specific, given adjectival property. In contrast, many of the above methods just create a figurative expression about the given concept without concern for a specific property. Second, the property is to be expressed indirectly via a metaphor, and the metaphor is further automatically generated for the given task. While the above methods often produce metaphoric expressions, they exercise less control over what the metaphor actually expresses. *Bislon* (Repar *et al.* 2018) is an exception: the user is expected to give a bisociated concept which could effectively act as a metaphorical vehicle. Additionally, in this paper we examine several internal evaluation functions used by our method, in order to gain insight into their value in generation of metaphorical slogans.



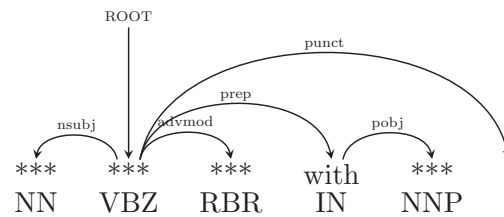


Fig. 1. An example of a skeleton constructed from Visa's slogan: "Life flows better with Visa."

#### 4. Method

Recall our goal: the user (or another software component, if this method is part of a larger system) specifies a concept  $T$  and a property  $P$ , and the method should suggest slogans related to  $T$ . These slogans should associate concept  $T$  with property  $P$ , preferably in a metaphoric manner, and use prosodic features and avoid negative expressions.

In a nutshell, the slogan generation process involves the following steps that will be detailed in the following subsections:

0. Construction of slogan skeletons, that is, slogan templates that have empty placeholders to be filled in with suitable words (Section 4.1).

Skeleton construction is performed just once to obtain a data set of skeletons for later use. All the following tasks are performed at runtime for the given concept  $T$  and property  $P$ .

1. *Metaphor generation* ( $((T, P) \mapsto v)$ ): Given a concept  $T$  and a property  $P$ , identify a suitable metaphorical vehicle  $v$  to associate the concept and property metaphorically (Section 4.2).
2. *Slogan search space definition* ( $((T, v, s) \mapsto \{\mathcal{E}_i\})$ ): Given a concept  $T$ , a vehicle  $v$  and a (random) skeleton  $s$ , identify sets of words that can potentially be used to fill in the placeholders in skeleton  $s$ , in order to obtain grammatical slogan expressions  $\mathcal{E}_i$  related to concept  $T$  and vehicle  $v$  (Section 4.3).
3. *Slogan filtering* ( $(\{\mathcal{E}_i\} \mapsto \{\mathcal{E}_j\} \subseteq \{\mathcal{E}_i\})$ ): Given candidate slogan expressions  $\mathcal{E}_i$ , filter out those lacking internal cohesion or with negative sentiment (Section 4.4).
4. *Internal slogan evaluation* ( $(f_d(T, P, \mathcal{E}_j) \rightarrow \mathbb{R})$ ): Given a concept  $T$ , a property  $P$  and a candidate slogan expression  $\mathcal{E}_j$ , evaluate the quality of the slogan along various dimensions  $f_d$  (Section 4.5).
5. *Finding good slogans*: Given the slogan search space  $\{\mathcal{E}_j\}$  and the internal evaluation dimensions  $f_d$ , carry out actual slogan expression generation and optimization to search the space for slogans  $\mathcal{E}_j$  with high  $f_d(\mathcal{E}_j)$  (Section 4.6).

##### 4.1. Construction of slogan skeletons

The slogan generation method reuses skeletons, that is, syntactical structures, extracted from existing slogans. Figure 1 shows the skeleton generated from Visa's slogan "Life flows better with Visa." Skeletons are to be filled in with appropriate words, so that a slogan results, as will be described in the following subsections.

A slogan skeleton is a parse tree of a sentence where content words are replaced with a placeholder "\*\*\*" and where grammatical relations between words and POS tags are maintained. A grammatical relation connects a word (called dependent) to its head word (called governor) with a specific type of relation. The POS tags are based on the Penn Treebank tag set (Marcus, Santorini, and Marcinkiewicz 1993).

Skeletons are constructed once and stored for later use in the slogan generation process. In order to construct a set of skeletons, for the experiments described in this paper we initially obtain 40 well-known good and modern slogans.<sup>e</sup>

We then manually preprocess the slogans to increase 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 misclassifications of verbs, adverbs, and adjectives as nouns (e.g., the adverb *differently* in *Red Lobster*'s slogan "Seafood Differently."). 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 the 40 slogans into parse trees using *spaCy* (Honnibal and Montani 2017). Skeleton candidates are obtained from the parse tree simply by keeping stop words but substituting all content words with placeholders ("\*\*\*\*"). Here we use stop words lists from *NLTK* (Bird, Klein, and Loper 2009).

We then keep only those skeletons that can be meaningfully used to generate novel slogans: an acceptable skeleton must have at least two placeholders, and the fraction of placeholders over all tokens must be at least 40%. These choices are made to avoid trivial adaptations, since slogans that are recognizable variations of other slogans are not likely to be good for branding. As a result, *Reebok*'s slogan "I am what I am." will not be reused: it contains no content words, only stop words, so the skeleton would be identical to the original slogan. Several slogans can also produce identical skeletons, for example, *Volkswagen*'s "Think Small." and *Apple*'s "Think Different." In total, the 40 slogans produce 26 unique skeletons (cf. Table 9).

#### 4.2. Computational generation of metaphors

The method aims to identify apt metaphorical vehicles that highlight a given property  $P$  in the given target concept (tenor)  $T$ . An example of such input is  $T = \text{computer}$  and  $P = \text{creative}$ . The vehicle identification step does not depend on the skeleton used.

The method begins by retrieving nouns associated with the input property  $P$  using two resources: *Thesaurus Rex* (Veale and Li 2013) is used to obtain general nouns such as *coffee* or *flower*, while the resource by Alnajjar *et al.* (2017) provides human categories such as *actor*, *lawyer*, or *politician*. The former will be used for generating general metaphors and the latter for personifications. Given a property, both resources provide a ranked list of nouns associated to this property. As the quality of their results vary, we empirically decided to use only the top 10% of each type, in order to obtain the nouns most strongly related to the given property  $P$ . These nouns are used as vehicle candidates.

For example, nouns most strongly associated with  $P = \text{creative}$  are  $\{\text{painting, music, } \dots, \text{presentation}\}$  and  $\{\text{artist, genius, poet, } \dots, \text{dancer}\}$  in the categories of general and personal nouns, respectively.

The vehicle candidates are not necessarily good vehicles, however, if they do not result in the intended metaphorical interpretation. We therefore use a separate metaphor interpretation model, *Meta4meaning* (Xiao *et al.* 2016), to assess the vehicle candidates in the context of tenor  $T$ .

*Meta4meaning* accepts two nouns as input, a tenor  $T$  and a (candidate) vehicle  $v$ , and produces a ranked list of possible interpretations for the corresponding nominal metaphor "[tenor]  $T$  is [vehicle]  $v$ ." In other words, *Meta4meaning* outputs a list of properties that it predicts the metaphor to assign to the tenor  $T$  via vehicle  $v$ . These are not necessarily the properties most strongly associated to vehicle  $v$  and they also depend on the tenor  $T$  (see later in this section).

<sup>e</sup> Retrieved from <http://www.advergize.com/advertising/40-best-advertising-slogans-modern-brands/2/> on 24 October 2016.

We keep a vehicle candidate  $v$  only if the desired property  $P$  is among the top 50 interpretations of the respective metaphor with tenor  $T$ . This ensures that the intended interpretation of the metaphor is feasible. The threshold of 50 top interpretations is chosen based on the results of Xiao *et al.* (2016), which indicate a recall of about 0.5 of human interpretations.

Our implementation of the *Meta4meaning* metaphor interpretation model (Xiao *et al.* 2016) uses the semantic model  $\omega$  described next to obtain measures of association between nouns and properties (based on word embedding). Following *Meta4meaning*, the method interprets the potential metaphors by considering the shared associations between the tenor and vehicle, and calculating the “combined metaphor rank” metric on them (cf. Xiao *et al.* 2016). In a nutshell, a property is considered a likely interpretation of the metaphor if either the property is strongly associated with both the tenor and the vehicle (as measured by the product of association strengths), or the property has a much stronger association to the vehicle than to the tenor. This metric highlights associations based both on semantic similarities and on salience imbalance between vehicle and tenor. Additionally, since metaphors are asymmetrical, we remove a vehicle candidate if the intended interpretation  $P$  is not better in the intended metaphor “ $T$  is [a]  $v$ ” than in the reverse metaphor, that is, “ $v$  is [a]  $T$ .”

Continuing our example, by interpreting all the vehicle candidates in the context of the tenor  $T = \text{computer}$  and keeping only those for which *creative* is among the top interpretations, we obtain vehicles  $\{\text{art, drama, . . . , exhibition}\}$  and  $\{\text{genius, artist, . . . , inventor}\}$  for the general and human categories, respectively. Finally, we merge the two lists of potential vehicles into one list.

To our knowledge, this proposed method is the first for generating metaphors based on their interpretations.

#### Semantic model $\omega$

We construct a simple semantic model in order to find words that are semantically related to a given word, and to measure the semantic relatedness between two given words. This semantic model is used in several parts of the slogan construction method, not just metaphor generation as described earlier in this section.

We follow the approach described for *Meta4meaning* (Xiao *et al.* 2016) in building the semantic model  $\omega$ . We obtain co-occurrence counts of words in *ukWaC*<sup>f</sup> (Baroni, Bernardini, Ferraresi, and Zanchetta 2009), a 2 billion word web-based text corpus. Co-occurrences are constrained by sentence boundaries and a window of  $\pm 4$  words. 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 a relatedness measure by employing the log-likelihood measure of Evert (2008) while capping all negative values to zero. Finally, we normalize relatedness scores using L1-norm following McGregor *et al.* (2015). As a result, an ambiguous word (e.g., *bank*) can be related to semantically different words (e.g., *money* and *river*). The semantic model does not aim to handle polysemy in any informed manner.

Examples of words related to the concept *computer* in the semantic model  $\omega$  include  $\{\text{system, software, network, skill, . . . , workstation}\}$ .

### 4.3. Search spaces for filling in skeletons

When producing slogan expressions, the method considers one skeleton  $s$  at a time, for the given concept  $T$  and vehicle  $v$ . The relationship to property  $P$  comes (metaphorically) via words related to vehicle  $v$ . Throughout this paper, we use vehicles generated by the metaphor generation process described above, but vehicle  $v$  could be input manually as well.

<sup>f</sup><http://wacky.sslmit.unibo.it>.

To instantiate a skeleton, the method constructs sets of words that can be used as potential fillers for each placeholder  $i$  in skeleton  $s$ . It starts by identifying the *grammatical space*  $\mathcal{G}_i$  consisting of all words that have the POS and grammatical relations matching placeholder  $i$  in skeleton  $s$ . Similar to the approaches by Özbal, Pighin, and Strapparava (2013) and Tomašič, Žnidaršič, and Papa (2014), we build a repository of grammatical relations, that is, of pairs of words that occur in each grammatical relationship to each other. The repository is built once, and is then used to identify  $\mathcal{G}_i$  at runtime by retrieving words that match the relevant relations from the repository. To construct the repository, we parse the entire *ukWaC* corpus using *spaCy* and store all grammatical relations observed along with their frequencies. We retain grammatical relations with frequencies at least 50 to remove rare and noisy cases. The process yields 3,178,649 grammatical relations, which are publicly available (Alnajjar 2018).

We then further identify those grammatical words that are also related either to the input concept  $T$  or the vehicle  $v$ , according to the semantic model  $\omega$  described above. This set of related and grammatical words is the *related space*  $\mathcal{R}_{i,T,v}$ , or just  $\mathcal{R}_i$  for short when the concept  $T$  and vehicle  $v$  are clear in the context. In order to identify the related space, the method obtains those words in  $\mathcal{G}_i$  that are either within the  $k$  words most strongly related to concept  $T$ , or within the  $k$  words most strongly related to vehicle  $v$ . In our case,  $k$  was empirically set to 150. Since abstraction tends to be required in processing metaphors (Glucksberg 2001), we only accept abstract terms related to vehicle  $v$ . For this, we utilize the abstractness data set provided by Turney *et al.* (2011) and keep words with abstractness level at least 0.5.

Given a skeleton  $s$ , concept  $T$  and vehicle  $v$ , the search space for possible slogans consists of all feasible ways of filling each placeholder  $i$  with a word from the respective related and grammatical space  $\mathcal{R}_i$ . Alternatively, if the above is not feasible, grammatical (unrelated) words in  $\mathcal{G}_i$  can be used as fillers.

As an example, let the skeleton  $s$  be  
 \*\*\*\_NN, \*\*\*\_NN and \*\*\*\_NN.

That is, three singular nouns (NN) separated by a comma and *and* (with grammatical relations omitted for simplicity). Let concept  $T$  be *computer* and vehicle  $v$  be *artist*. The grammatical space  $\mathcal{G}_{i=1}$  for the first placeholder consists of all singular nouns in the grammatical repository (that satisfy all relations linked to it, such as the “punc” relation to the second token “,”). Examples of filler candidates in  $\mathcal{G}_1$  are {*management, talent, site, skill, . . . , health*}. The related and grammatical space  $\mathcal{R}_1$  for the same placeholder is the subset of  $\mathcal{G}_1$  that is related to *computer* or *artist* in the semantic model  $\omega$ : {*system, skill, programming, art, designer, talent, simulation, . . .*}. A random filler word is then selected from  $\mathcal{R}_1$  (e.g., *talent*) or, if the set were empty, then an (unrelated) filler is chosen at random from  $\mathcal{G}_i$ . This process is repeated for each placeholder, yielding slogans such as

“software, design and simulation.”

and

“talent, talent and support.”

#### 4.4. Filtering criteria for slogan expressions

Not all expressions in the search space defined above are suitable as slogans. We use two criteria to filter out expressions that are not likely to be good slogans: lack of cohesion within the expression, and negative sentiment.

*Semantic cohesion* is measured to avoid slogans that have mutually unrelated words. We require that all content words (i.e., words used in the placeholders) are semantically related to each other, according to the semantic model  $\omega$ . If any pair of content words is not related, the expression is

discarded. Alternatively, we could use a nonbinary measure of cohesion. We will return to this in the discussion.

As advertising slogans tend to be positive expressions (Dowling and Kabanoff 1996), we employ *sentiment analysis* to prevent negative sentiment. We use the sentiment classifier provided in *Pattern* (De Smedt and Daelemans 2012) to predict the sentiment polarity score of expressions. The score is a value between  $-1$  and  $+1$ ; we discard slogan expressions with a negative score.

#### 4.5. Internal evaluation dimensions for slogan expressions

With the spaces  $\mathcal{R}_i$  and  $\mathcal{G}_i$  and the filtering criteria above, we have defined a space of possible slogans. Still, some expressions in the space are likely to be better slogans than others, and we next define four internal evaluation dimensions that the slogan generator can use. Our hypothesis is that the dimensions are useful ones, and we will test this hypothesis empirically in the experimental section.

The four dimensions are (1) target relatedness, that is, relatedness to concept  $T$  and property  $P$ , (2) language, (3) metaphoricity, and (4) prosody. Each dimension can be further composed of multiple sub-features.

##### 4.5.1. Target relatedness (to concept $T$ and property $P$ )

Slogan expressions generated according to the above-defined constraints relate to concept  $T$  and property  $P$  to varying degrees. By construction, the search space favors content words that are related to concept  $T$  or vehicle  $v$ , but property  $P$  is not considered directly because we want to encourage this relation to be metaphoric. Given that a slogan eventually intends to connect property  $P$  to concept  $T$ , it seems natural to measure and possibly maximize the relationship of the slogan expression to the target input, that is, both concept  $T$  and property  $P$ .

Formally, we measure semantic relatedness  $f_{rel}(\mathcal{E}, w)$  between a slogan expression  $\mathcal{E}$  and a single target word  $w$  as the mean relatedness

$$f_{rel}(\mathcal{E}, w) = \frac{\sum_{t \in c(\mathcal{E})} \omega(t, w)}{|c(\mathcal{E})|} \quad (1)$$

where  $c(\mathcal{E})$  is the set of content words (i.e., filler words in placeholders) in slogan expression  $\mathcal{E}$  and  $\omega(t, w)$  is a score given by the semantic relatedness model  $\omega$ . The internal evaluation dimension of *relatedness* (to concept  $T$  and property  $P$ ) is computed as a weighted sum of the semantic relatedness of the slogan expression to  $T$  and to  $P$ . The weights are given in Table 1. (The other three dimensions are also computed as weighted sums of their sub-features; all weights are given in the table.) We chose to give relatedness to  $P$  a higher weight as the search space already consists of words related to the concept  $T$ .

##### 4.5.2. Language

Skeletons, with their grammatical relations and POS tags, aim to ensure that slogan expressions produced with them are likely to be grammatically correct. However, these constraints are not sufficient to guarantee correctness. We resort to a simple statistical method, bigrams, to obtain an alternative judgment, in the form of a likelihood of the slogan expression in comparison to a large corpus. In addition, under the language dimension, we also consider surprisingness (rarity) of the individual words in the expression.

We build a probabilistic language model using bigram frequencies provided with the *ukWaC* corpus. A slogan with higher probability according to the language model is more likely to be grammatically correct as its bigrams appear more frequently in the *ukWaC* corpus. Employing

**Table 1.** The weights assigned to each sub-feature in the four internal evaluation dimensions

| Dimension                  | Feature                                 | Weight |
|----------------------------|---|--------|
| Relatedness                | $f_{rel}(\mathcal{E}, T)$               | 0.4    |
|                            | $f_{rel}(\mathcal{E}, P)$               | 0.6    |
| Language                   | $Prob(\mathcal{E})$                     | 0.8    |
|                            | $f_{unusual}(\mathcal{E})$              | 0.2    |
| Metaphoricity <sup>a</sup> | $f_{metaph-maxrel}(\mathcal{E}, T, v)$  | 0.5    |
|                            | $f_{metaph-diffrel}(\mathcal{E}, T, v)$ | 0.5    |
| Prosody                    | $f_{rhyme}(\mathcal{E})$                | 0.4    |
|                            | $f_{alliteration}(\mathcal{E})$         | 0.4    |
|                            | $f_{assonance}(\mathcal{E})$            | 0.1    |
|                            | $f_{consonance}(\mathcal{E})$           | 0.1    |

<sup>a</sup>In case the value of this dimension is negative (i.e., when a word in the expression  $\mathcal{E}$  is related to the concept/tenor  $T$  more than to the metaphorical vehicle  $v$ ), it is capped to zero.

bigrams, in contrast to trigrams or higher  $n$ -grams, gives the method a greater degree of freedom in its generation; higher  $n$ -grams would improve the grammar of the generated expressions but would tie them to expressions in the original corpus.

Surprisingness is the other feature we consider in the language dimension, inspired by Özbal, Pighin, and Strapparava (2013). We measure how infrequent, that is, unusual, the individual words in the slogan are

$$f_{unusual}(\mathcal{E}) = \frac{\sum_{t \in c(\mathcal{E})} \frac{1}{freq(t)}}{|c(\mathcal{E})|} \quad (2)$$

where  $freq(t)$  is the absolute frequency of word  $t$  in the *ukWaC* corpus, and where word  $t$  is ignored in the computation (both nominator and denominator) if its frequency  $freq(t)$  is zero. While such words could be surprising, they also add noise, so we consider it safer to ignore them. In case no content word appears in the corpus, the surprisingness score is defined to be zero; that is, we conservatively consider the expression not to be surprising. The weights assigned to these sub-features when representing the entire language dimension were set empirically (cf. Table 1).

#### 4.5.3. Metaphoricity

By construction, slogan expressions in the defined search space are encouraged to be metaphorical, but their degree of metaphoricity varies. We define two functions that aim to measure some aspects of metaphoricity in the produced slogan expressions. In these functions, we use both the concept/tenor  $T$  and the metaphorical vehicle  $v$  used in the construction of the expression.

The first function,  $f_{metaph-maxrel}$ , considers the strongest relationships between any of the content words  $t \in c(\mathcal{E})$  in slogan  $\mathcal{E}$ , and the tenor  $T$  and the vehicle  $v$ :

$$maxrel(\mathcal{E}, w) = \max_{t \in c(\mathcal{E})} \omega(t, w) \quad (3a)$$

$$f_{metaph-maxrel}(\mathcal{E}, T, v) = maxrel(\mathcal{E}, T) \cdot maxrel(\mathcal{E}, v) \quad (3b)$$



where  $\omega(\cdot)$  is a score given by the semantic relatedness model. When this function has a value larger than zero, then the slogan contains a word that is related to the concept/tenor and a (possibly same) word that is related to the vehicle. The larger the value, the more related these words are to the concept/tenor and vehicle. Obviously, a slogan that is not (strongly) related to both the concept  $T$  and the vehicle  $v$  can hardly be metaphorical in the intended manner.

The other metaphoricity function,  $f_{\text{metaph-diffrel}}$ , checks whether the slogan expression  $\mathcal{E}$  contains a word  $t$  that is strongly related to the metaphorical vehicle  $v$  but *not* to the concept/tenor  $T$ . The hypothesis is that such a word  $t$  is more likely to force a metaphorical interpretation of the expression, in order to connect  $t$  to the concept/tenor  $T$ . For instance, let the tenor  $T$  be *car* and the vehicle  $v$  be *dancer*, and let candidate content words related to *dancer* be *stage* and *street*. The expression “cars of stage” is much more likely to have a metaphorical interpretation than the expression “cars of street,” since the word *stage* used in the former is *not* related to cars. Function  $f_{\text{metaph-diffrel}}$  is introduced to measure and encourage this metaphoricity arising from words  $t$  related to the vehicle  $v$  but *not* to the concept/tenor  $T$  as follows:

$$f_{\text{metaph-diffrel}}(\mathcal{E}, T, v) = \max_{t \in c(\mathcal{E})} (\omega(t, v) - \omega(t, T)) \quad (4)$$

The internal dimension of metaphoricity is obtained as the sum of the two sub-features, that is, they are given equal importance.

#### 4.5.4. Prosody

In our work, we consider four features of *prosody*: rhyme, alliteration, assonance, and consonance. For this, we make use of *The CMU Pronouncing Dictionary* (Lenzo 1998) to analyze repeated sounds in words. *The CMU Pronouncing Dictionary* is a mapping dictionary from English words to their phonetic translations. While the dictionary is limited by its vocabulary, the vocabulary is relatively extensive as it contains over 134,000 words.

Let  $\varphi(t)$  be CMU’s function which returns the sequence of phonemes in a given text (word  $t$  or slogan  $\mathcal{E}$ ), and let *vowels* be the set of (phonetic transcriptions of) vowels.  $\mathbb{1}_X$  is an indicator function that returns 1 if  $X$  is true and 0 otherwise.

Equation (5a) is for counting the total number of occurrences of phoneme *pho* in slogan  $\mathcal{E}$ . We only consider sounds repeated at least three times (Equation (5b)).

$$\text{count}_{\text{phoneme}}(\mathcal{E}, \text{pho}) = \sum_{t \in \mathcal{E}} \sum_{p \in \varphi(t)} \mathbb{1}_{p=\text{pho}} \quad (5a)$$

$$\text{count}_{\text{phoneme} \geq 3}(\mathcal{E}, \text{pho}) = \begin{cases} \text{count}_{\text{phoneme}}(\mathcal{E}, \text{pho}), & \text{if } \text{count}_{\text{phoneme}}(\mathcal{E}, \text{pho}) \geq 3 \\ 0, & \text{otherwise} \end{cases} \quad (5b)$$

We implement the *assonance* and *consonance* functions by considering the total relative frequency of vowels or consonants, respectively, that are repeated at least three times:

$$f_{\text{assonance}}(\mathcal{E}) = \frac{\sum_{\text{pho} \in \text{vowels}} \text{count}_{\text{phoneme} \geq 3}(\mathcal{E}, \text{pho})}{|\{\varphi(\mathcal{E})\}|} \quad (6a)$$

$$f_{\text{consonance}}(\mathcal{E}) = \frac{\sum_{\text{pho} \notin \text{vowels}} \text{count}_{\text{phoneme} \geq 3}(\mathcal{E}, \text{pho})}{|\{\varphi(\mathcal{E})\}|} \quad (6b)$$

For *alliteration* and *rhyme*, we count the number of word pairs that share their first or last phonemes, respectively, regardless of their quality and stress. For simplicity, syllables are not taken into account. Denoting the first phoneme in a word  $t$  by  $\varphi(t)_0$  and the last by  $\varphi(t)_{-1}$ , the measures are as follows:

$$f_{alliteration}(\mathcal{E}) = \frac{\sum_{t_i, t_j \in \mathcal{E}, t_i \neq t_j} \mathbb{1}_{\varphi(t_i)_0 = \varphi(t_j)_0}}{|\mathcal{E}|} \quad (7a)$$

$$f_{rhyme}(\mathcal{E}) = \frac{\sum_{t_i, t_j \in \mathcal{E}, t_i \neq t_j} \mathbb{1}_{\varphi(t_i)_{-1} = \varphi(t_j)_{-1}}}{|\mathcal{E}|} \quad (7b)$$

#### 4.6. Algorithm for finding good slogans

We employ genetic algorithms to find good slogans in the above-described space of possible expressions, given a skeleton  $s$ , related words  $\mathcal{R}_i$ , and grammatical words  $\mathcal{G}_i$  for each placeholder  $i$ , as well as the filtering criteria and internal evaluation dimensions described above. We use Deap (Fortin *et al.* 2012) as the evolutionary computation framework. Next, we use  $\mu$  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.

As an overview, the algorithm first produces an initial population of slogan expressions (“individuals”) and then evolves it over  $G$  iterations. Starting with the initial population, the employed  $(\mu + \lambda)$  evolutionary algorithm produces  $\lambda$  number of offspring by performing crossovers and mutations according to the respective probabilities  $Prob_m$  and  $Prob_c$ . The algorithm then puts the current population and offspring through a filtering process (described below). The population for the next generation is produced by evaluating the current population and the offspring, and then selecting  $\mu$  number of individuals. The evolutionary process ends after the specified number of generations. Details of the process will be given in the following paragraphs.

*Initial population.* Given a skeleton  $s$ , related words  $\mathcal{R}_i$ , and grammatical words  $\mathcal{G}_i$ , the algorithm produces a new individual (i.e., slogan expression) as follows. It begins by filling the placeholder with the most dependent words to it, usually the root. The algorithm attempts to randomly pick a related word from  $\mathcal{R}_i$ . If, however, the set is empty, that is, there are no related and grammatical words that can be used in the placeholder, a grammatical word is randomly picked from the set  $\mathcal{G}_i$ . The algorithm repeats the above steps to fill in the rest of the placeholders, always taking into account the conditions imposed by the already filled words. If the method fails to locate a suitable filler for a placeholder also in  $\mathcal{G}_i$ , the individual (expression) is discarded and the filling process starts over with a new individual. The process above is repeated until the desired number of individual expressions is generated, serving as the initial population.

*Mutation, crossover, and filtering.* Our algorithm employs one type of mutation which substitutes filler words in placeholders. The probability of producing an offspring by mutation is  $Prob_m$ . In the substitution, the mutation operation follows a similar process as for the initial population to find a related and grammatical word for the placeholder. For instance, mutating the slogan

“talent, talent and support.”

begins by turning a random content word back into a placeholder (e.g., “talent, \*\*\* \_NN and support.”) and then filling the placeholder with a new word from the relevant space  $\mathcal{R}_i$ . A new variant of the slogan results, such as

“talent, design and support.”

The algorithm applies a one-point crossover on two individuals with probability  $Prob_c$ ; that is, any pair of individuals is crossed over with probability  $Prob_c$ . As an example, a crossover of the two slogans

“work, skill and inspiration.”

“talent, design and support.”

after the third token would yield

“work, skill and support.”

“talent, design and inspiration.”

The resultant newly generated child expressions are put through a grammatical check, verifying that the filler word in each placeholder  $i$  is in the grammatical space  $\mathcal{G}_i$  also when considering the other content words that may have changed meanwhile. A failure of the grammatical check, for any of the two children, results in their disposal while parent expressions are kept in the population.

All offspring are filtered based on lack of internal cohesion, or negative sentiment, as described in Section 4.4. Additionally, mutation and crossover may produce duplicate slogans; once a new generation is produced, the filtering process also removes any duplicates.

*Fitness functions and selection.* The genetic algorithm uses the four internal evaluation dimensions defined in Section 4.5 as its fitness functions: (1) target relatedness, (2) language, (3) metaphoricity, and (4) prosody.

Some of the evaluation dimensions are conflicting in nature. For instance, the target relatedness dimension favors words related to the target concept  $T$  and property  $P$ , while the metaphoricity dimension favors words related to concept  $T$  and the metaphorical vehicle  $v$ . A single ranking method for selection, based on some linear combination of the dimensions, would not allow different trade-offs between the evaluation dimensions. Instead, our selection process involves the nondominant Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm (Deb *et al.* 2002) that looks for Pareto-optimal solutions, that is, solutions that cannot be improved any further without degrading at least one of the internal evaluation dimensions. This approach supports diversity among multiple, potentially conflicting objectives.

## 5. Empirical evaluation

We carried out human evaluations of both slogans and metaphors generated by the method. Metaphors were evaluated on their own since their generation method is novel, and since metaphors have a central role in the slogan generation process. The evaluations were carried out as crowdsourced surveys on Crowdfunder.<sup>8</sup> Crowdsourcing allowed us to gather large amounts of judgments of metaphors and slogans and to carry out quantitative analysis on them. We targeted our surveys to the following English-speaking countries: the United States, the United Kingdom, New Zealand, Ireland, Canada, and Australia.

As input to the slogan generation system, we used concept–property pairs and let the system generate slogans for them. Given that the space of possible concept–property pairs for slogan generation is not closed, and that no obvious distribution exists from which to draw a representative sample of concept–property pairs, we resorted to manually selecting a diverse collection of 35 concept–property pairs (Table 2). These pairs were inspired by Xiao and Blat (2013) and defined by the authors of this paper to represent a range of different concepts and different properties, including both typical (“chocolate is sweet”) and less typical associations (“computer is creative”). The aim is to use this set as a proof of concept across a range of slogan generation tasks; the results obviously are specific to this data set. The concept–property pairs were chosen before the tests described in the following were carried out, so they have not been cherry-picked to give good results. From the 35 concept–property pairs, we generated 212 metaphors and subsequently 684 slogans. Each slogan and metaphor was evaluated through Crowdfunder.

<sup>8</sup> [www.crowdfunder.com](http://www.crowdfunder.com).

**Table 2.** The 35 concept–property pairs used to evaluate the methods

| 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 |

Each property is used individually with the respective concept.

By design, a main goal of the slogan generation method proposed in this paper is to produce metaphoric slogans. Given the central role of metaphors for the method, we first evaluate the metaphor generation component. Discussion of the results is deferred to Section 6.

### 5.1. Evaluation of metaphor generation

As described in Section 4.2, the metaphor generation method is based on a metaphor interpretation model; that is, the method looks for an apt vehicle such that the interpretation of the resulting metaphor is as close to the intended meaning as possible. In this evaluation, we compare these generated apt vehicles to various baselines.

Given the 35 inputs in Table 2, the method produced 53 apt vehicles, that is, vehicles that are considered by the method to highlight the input property  $P$  in the input concept/tenor  $T$ . Out of these vehicles, 31 are general nouns and 22 are human. Tables 3 and 4 list ten random examples of generated vehicles in both classes, respectively (column “Generated Apt Vehicles”).

For each generated apt vehicle, we generated three matching baseline vehicles without the metaphor interpretation model:

- A *strongly related* vehicle is selected at random among the same top 10% of nouns associated to property  $P$  as considered by the metaphor generation method (cf. Section 4.2), but under the constraint that it is not considered apt by the generation method.
- A *related* vehicle is selected randomly among the bottom 90% of nouns associated with property  $P$ .
- A *random* vehicle is picked from those nouns that are not associated at all with property  $P$ .

Given that we have two classes of vehicles, general and human, we picked the baseline vehicles always from the same class as the apt vehicle. Baseline vehicles for the random examples are also given in Tables 3 and 4.

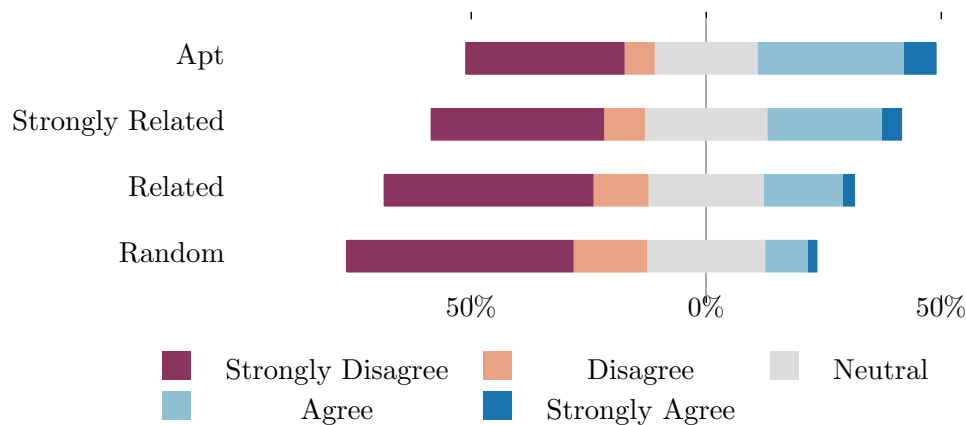
**Table 3.** Random examples of vehicles in the class of general nouns, both the apt vehicle generated by the method and three baseline vehicles

| Input      |            | Generated    | Baselines        |               |             |
|------------|------------|--------------|------------------|---------------|-------------|
| Tenor      | Property   | Apt vehicle  | Strongly related | Related       | Random      |
| book       | valuable   | purse        | image            | ginger        | metal       |
| painting   | elegant    | velvet       | tuberoses        | aluminum      | gps         |
| car        | elegant    | scarf        | tuberoses        | mahogany      | mold        |
| professor  | smart      | refrigerator | dolphin          | weapon        | pomfret     |
| computer   | creative   | poet         | performance      | speech        | bittersweet |
| professor  | old        | tractor      | printer          | beads         | timber      |
| politician | Aggressive | bullying     | wrestling        | skateboarding | ambulance   |
| chocolate  | Healthy    | colon        | herb             | aorta         | tantrism    |
| museum     | Ancient    | latin        | brachiopod       | universe      | crocodile   |
| love       | beautiful  | art          | line             | moonstone     | deerskin    |

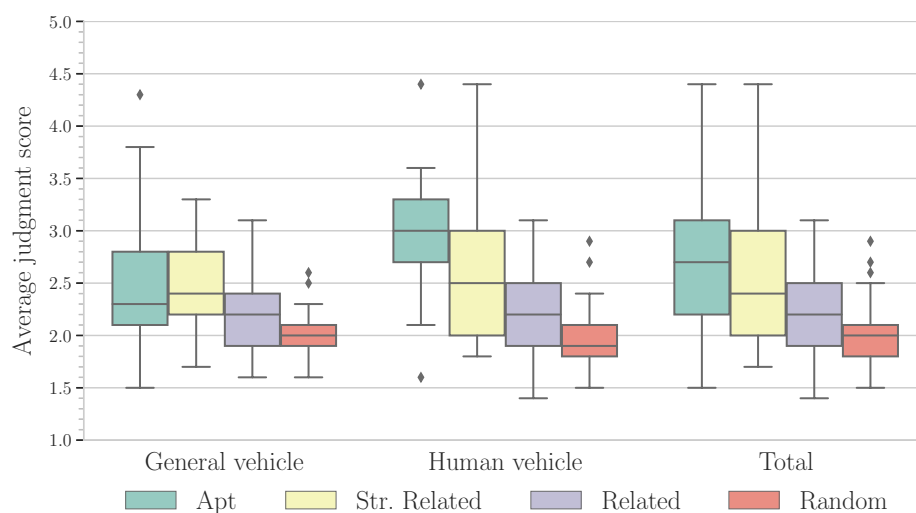
**Table 4.** Random examples of vehicles in the class of humans, both the apt vehicle generated by the method and three baseline vehicles

| Input      |            | Generated   | Baselines        |              |                 |
|------------|------------|-------------|------------------|--------------|-----------------|
| Tenor      | Property   | Apt vehicle | Strongly related | Related      | Random          |
| book       | wise       | father      | judge            | brother      | marker          |
| museum     | scientific | scientist   | computer         | technologist | apartment       |
| computer   | powerful   | king        | tyrant           | mogul        | grief           |
| politician | powerful   | monster     | emperor          | thug         | temple          |
| professor  | wise       | king        | father           | politician   | executive       |
| coke       | sweet      | mother      | friend           | mistress     | cinema          |
| coke       | dark       | demon       | terrorist        | spy          | travel          |
| paper      | scientific | scientist   | computer         | philosopher  | hexachlorophene |
| professor  | old        | child       | king             | invalid      | tendon          |
| love       | wild       | cat         | warrior          | pirate       | orator          |

Given the 53 generated apt vehicles and three baselines for each of them, we obtained a total of 212 metaphors to evaluate. For the evaluation, we represented 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 five-point Likert scale where 1 indicates strong disagreement and 5 strong agreement. The order of metaphors was randomized for each judge. Ten judges were required to evaluate every metaphor.



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



**Fig. 3.** Distributions of mean judgements over metaphors with different types of vehicles (apt vehicles used by the method, strongly related baseline, related baseline, and random baseline). Results are given separately for general and human classes of vehicles, as well as for their combination ("Total"). Plots indicate the median, first and third quartiles and 95% intervals.

A summary of results is given in Figure 2 in the form of a diverging bar chart illustrating the percentages of judgments on the Likert scale for each type of vehicles tested (the generated apt vehicle, and the baselines of strongly related, related, and random).

We can observe that apt vehicles performed best, followed by the baseline vehicles in the order of strength of relatedness to the property. 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 3 shows the distributions of mean judgements over the metaphors generated. The first group of bars is for metaphors with general vehicles, the second group with human vehicles, and the third group represents their union. Table 5 provides the respective numbers.

Based on the results, we can observe that apt and also strongly related vehicles of the human class performed best. Their median scores of 3.0 and 2.5, respectively, also outperform apt general vehicles (median 2.3). Within the group of general vehicles, apt and strongly related vehicles performed best.



**Table 5.** Five-number summaries (median, first and third quartiles, minimum and maximum values) of the mean judgments of metaphors

|              | Generated   |     |              |     | Baselines |     |        |     |
|--------------|-------------|-----|--------------|-----|-----------|-----|--------|-----|
|              | Apt vehicle |     | Str. related |     | Related   |     | Random |     |
| General      | 2.3         |     | 2.4          |     | 2.2       |     | 2.0    |     |
| vehicles     | 2.1         | 2.8 | 2.2          | 2.8 | 1.9       | 2.4 | 1.9    | 2.1 |
| ( $n = 31$ ) | 1.5         | 4.3 | 1.7          | 3.3 | 1.6       | 3.1 | 1.6    | 2.6 |
| Human        | 3.0         |     | 2.5          |     | 2.2       |     | 1.9    |     |
| vehicles     | 2.8         | 3.3 | 2.0          | 3.0 | 1.9       | 2.5 | 1.8    | 2.1 |
| ( $n = 22$ ) | 1.6         | 4.4 | 1.8          | 4.4 | 1.4       | 3.1 | 1.5    | 2.9 |
| Total        | 2.7         |     | 2.4          |     | 2.2       |     | 2.0    |     |
| ( $n = 53$ ) | 2.2         | 3.1 | 2.0          | 3.0 | 1.9       | 2.5 | 1.8    | 2.1 |
|              | 1.5         | 4.4 | 1.7          | 4.4 | 1.4       | 3.1 | 1.5    | 2.9 |

$n$  denotes the number of metaphors evaluated; the number of individual judgments is tenfold.

The combined results (group “Total”) suggest that the generated apt vehicles outperform the baselines. A statistical test validates this observation. Nonparametric permutation test shows that the mean judgment of apt vehicles is statistically significantly higher than the mean judgment of strongly related vehicles,  $P = 0.0074$  (one-tailed).

## 5.2. Evaluation methodology for slogan generation

We next evaluate the generated slogans. The primary goal is to identify whether the proposed method is capable of producing expressions suitable for the task, that is, feasible as advertising slogans. A secondary goal is to investigate the effects of the evaluation dimensions of the genetic algorithm on the produced slogans. With this, we hope to shed light on computational criteria for future slogan generation methods. The evaluation setup for slogan generation is the following.

For every triplet of concept  $T$ , property  $P$ , and (apt) vehicle  $v$  obtained from the metaphor generation stage, we randomly select two skeletons. In our experiments, we have a set of 26 skeletons to choose from; the number of skeletons applied per input is here limited to two for simplicity of experimental design. In real applications, a wider selection would provide more variation.

One skeleton at a time is filled in by the genetic algorithm. We empirically set the following values for parameters of the genetic algorithm:  $\mu = \lambda = 100$ ,  $G = 25$ ,  $Prob_c = 0.4$ ,  $Prob_m = 0.6$ .

We selected multiple slogans for evaluation from the final population produced by the genetic algorithm, in order to study the effects of various evaluation dimensions on the quality of slogans. As described in Section 4.5, there are four **internal evaluation dimensions**: (1) *relatedness* of the slogan to the concept and the property given as input, (2) *language*, (3) *metaphoricity*, and (4) *prosody*. Because these dimensions are partially mutually contradictory, we evaluate slogans that have different trade-offs between them. For the experiments of this paper, we used three **selection methods** for slogans:

- *Balanced* dimensions: A randomly selected slogan that has a positive value on several internal evaluation dimensions. In addition to requiring that all four dimensions are positive, we also try the cases where this requirement is relaxed either for prosody or for metaphoricity.

- A *maximized* dimension: A slogan with the maximum value on one of the four dimensions, regardless of other dimensions.
- *Minimized* dimensions: A random slogan with the lowest values on all four dimensions (relatedness, language, metaphoricity, and prosody, considered in order).

This selection yielded 684 slogans to be evaluated. The balanced selection failed for some cases because no slogan in the generated population met the selection criteria.

In order to represent the slogans in a uniform, slogan-like style, we detokenize them using *NLTK*, capitalize the words in them, and add a full stop in the end.

We asked five judges to evaluate each selected slogan on a five-point Likert scale based on the following five aspects or **judgments**: (1) the *relatedness* of the slogan to the title (i.e., input concept and property), (2) the *language correctness*, (3) the *metaphoricity*, (4) the *catchiness*, *attractiveness* and *memorability*, and (5) the *overall quality* of the expression as a slogan.

These judgments and the internal evaluation dimensions described above consider similar aspects. With this design, we intend to measure how well the internal evaluation dimensions are reflected in the output, as well as to test how they contribute to the overall quality of the generated slogans.

To simplify some of the analyses next, we consider the overall quality of an individual slogan to be *good* if the mean judgment is above 3 for the question “Overall, this is a good slogan.” In some of the analyses, we also do such dichotomization to the other judgments.

For a comparison of computer-generated slogans to professionally crafted ones, we ran a similar survey with slogans produced by professionals for past advertising campaigns. We use <http://www.textart.ru/><sup>h</sup> due to its consistent structure of listing slogans and wide coverage of slogans from different categories. The corpus includes additional information regarding slogans such as the name of the brand and its category (e.g., pizza or university). In the experiment, we use 100 random slogans obtained from the above site. In order to reduce the effect of familiarity of the brand on the evaluation, we manually substituted product and brand names with the text “ProductName.” We also had to adjust the first evaluation question about relatedness: due to the lack of explicit input concepts and properties in the human-made slogans, we used the product’s category (provided in the database) as the target concept *T* and removed the property *P* from the question. We required 10 judges to provide their opinions on each human-made slogan and thus received a total of 1000 judgments.

It is worth noting that a direct comparison between the results of computer-made slogans and human-made ones is not feasible. First, the two evaluations are not identical (e.g., missing the adjectival properties from evaluated human-made slogans, nonequivalent number of judges, and nonidentical judges); second, some artificial constraints were enforced during computational slogan production (e.g., computer-made slogans were restricted to two skeletons). It is also good to keep in mind that generated slogans are intended as slogan candidates for brainstorming. Nevertheless, juxtaposing the results for computer-generated and existing slogans can give useful insights.

### 5.3. Overview of results for slogan generation

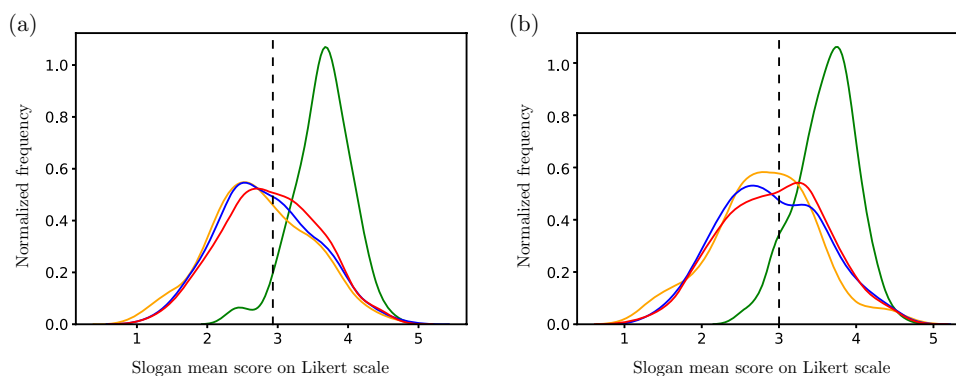
As concrete examples of what the experimental setup produced, Table 6 shows some generated slogans, both more and less successful ones.

Figure 4 gives the distributions of judgments on the overall suitability of slogans, and on their catchiness. Slogans created by professionals stand out, as expected, but the generated slogans fair well, too. The judgments are centered around 3 and have a relatively wide distribution, indicating

<sup>h</sup> Collected on 24 October 2016.

**Table 6.** Examples of generated slogans

| Concept    | Property   | Vehicle    | Output                                  |
|------------|------------|------------|---|
| computer   | creative   | artist     | "Talent, Skill And Support."            |
| computer   | creative   | artist     | "Follow Questions. Start Support."      |
| computer   | creative   | poet       | "Work Unsupervised."                    |
| computer   | creative   | poet       | "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." |
| politician | dishonest  | thief      | "Free Speech."                          |
| politician | aggressive | predator   | "Media For A Potential Attack."         |



**Fig. 4.** Distributions of judgments for overall quality and catchiness for generated slogans (*balanced* in red, *maximized* in blue, and *minimized* in orange) and expert-written slogans (in green). (The graphs show distributions over slogans, where each slogan is represented by its mean score.). (a) Overall quality. (b) Catchiness.

that while most slogans are neutral in the Likert scale, there are also some relatively good and some relatively poor ones.

A comparison between different selection methods indicates that balanced slogans contain somewhat more suitable ones (i.e., with scores larger than 3) than the other selection methods. This observation is similar for catchiness (Figure 4(b)) and for other judgments (not shown).

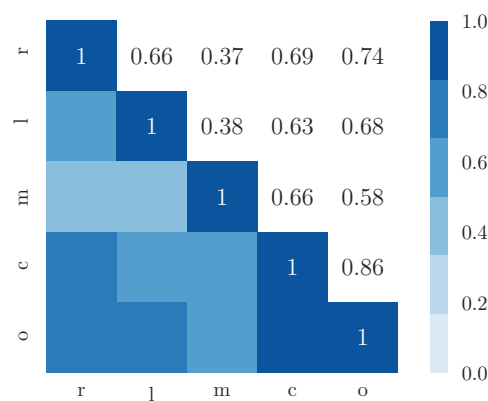
Table 7 provides a numerical summary of the performance of slogans with regard to all judgments. We observe that the balanced selection performs best in all judgments and the minimized selection worst by a clear margin. A comparison across different judgments in Table 7 shows that language correctness received the best scores, followed by relatedness, catchiness, and, finally, metaphoricality.

In total, over all selection methods, 35% of generated slogans were judged to be suitable (and 39% of the balanced slogans). The input that resulted in most suitable slogans was *computer-powerful*, with 13 suitable slogans out of 20 generated for it. On the other hand, input *newspaper-international* had the least number of successful slogans, 1 out of 12. This means that the method has generated at least one successful slogan for each input, even though we only used two random skeletons for each input.

**Table 7.** The percentage of slogans being judged as successful with respect to different aspects

| Selection method        | Relatedness (%) | Language (%) | Metaphoricity (%) | Catchiness (%) | Overall (%) |
|-------------------------|-----------------|--------------|-------------------|----------------|-------------|
| Balanced ( $n = 466$ )  | 48              | 52           | 39                | 44             | 39          |
| Maximized ( $n = 389$ ) | 45              | 49           | 39                | 40             | 35          |
| Minimized ( $n = 104$ ) | 28              | 38           | 28                | 36             | 32          |
| Expert ( $n = 100$ )    | 94              | 98           | 84                | 89             | 92          |

A slogan is considered successful if the respective mean score is greater than 3.

**Fig. 5.** Pearson correlation coefficient of judgments on human-made slogans between the five questions: (r)elatedness, (l)anguage, (m)etaphoricity, (c)atchiness, and (o)verall quality.

#### 5.4. Human judgments and evaluation dimensions

In this paper, we decided to focus on four different aspects of slogans: relatedness, language (correctness), metaphoricity, and catchiness/prosody. How do these four aspects relate to the overall suitability of slogans?

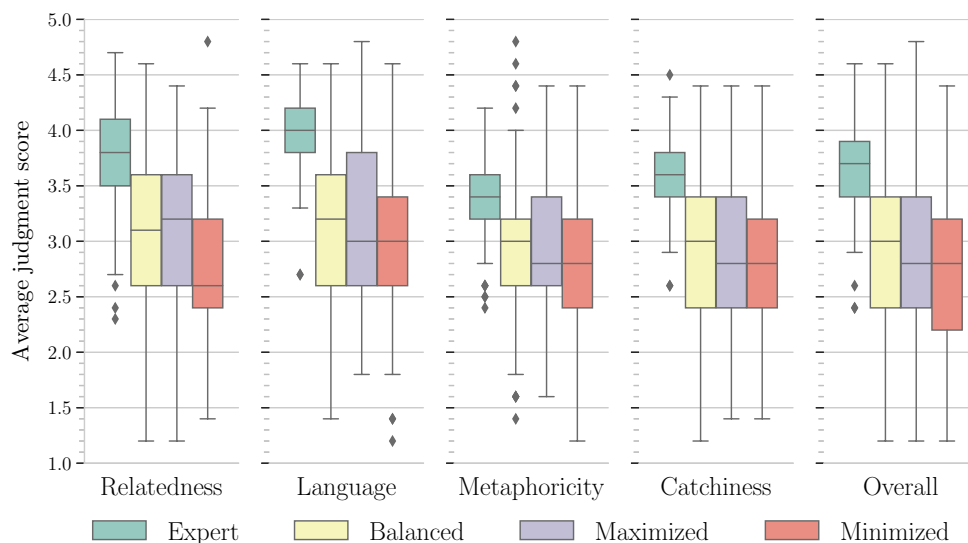
We measured all correlations between the four human judgments and the overall quality using human-made slogans (Figure 5). Correlations of the four judgments with the overall quality are strong (line and column “o” in the figure), ranging from 0.86 for catchiness to 0.58 for metaphoricity. This suggests that all four aspects contribute to the overall quality of slogans, especially catchiness and relatedness.

Correlations between the four judgments tend to be strong as well, over 0.5, except for correlations between metaphoricity and relatedness (0.37), and between metaphoricity and language correctness (0.38).

Overall, the high levels of correlation between the four judgments and the overall suitability suggest that all the four aspects should be balanced rather than only maximizing some of them. This is in line with the observation made above that a balanced selection produces better slogans.

*Human judgments versus internal evaluation dimensions.* Above we established that catchiness, relatedness, language, and metaphoricity are all important factors in slogans. How well do the respective internal evaluation dimensions correlate with the judgments in the survey, that is, does the method optimize the right things?

Here, we consider the sets of successful and unsuccessful slogans with respect to each human judgment type separately, and compute the mean values of the corresponding internal evaluation dimension in both sets.



**Fig. 6.** Distributions of mean judgments of slogans, for expert-written as well generated ones with different selection methods (balanced, maximized, or minimized internal dimensions). Results are given separately for different human judgments (relatedness, language, metaphoricity, catchiness, and overall quality). For each judgment, the “maximized” results shown are for the case where the corresponding internal evaluation dimension was maximized by the method; the “overall” case is their aggregation. Plots indicate the median, first and third quartiles, and 95% intervals.

Permutation tests indicate statistically significant associations between the internal evaluation dimensions and the respective human judgments for relatedness ( $P = 10^{-6}$ ), for metaphoricity ( $P = 0.0033$ ), and for prosody/catchiness ( $P = 0.046$ ), but not for language (correctness) ( $P = 0.84$ ).

### 5.5. Different slogan selection criteria

We next consider the different selection methods (balanced, maximized, or minimized internal evaluation dimensions) as well as different human judgments of the respective slogans. An overview of the results is given in Figure 6 while more details are available in Table 8. The general overview corresponds to the observations above.

Looking at the overall judgments (last group on the right in Figure 6), we notice—as before—that slogans with balanced dimensions tend to be appreciated more than slogans with a single maximized dimension. The first four groups look at the four specific human judgments, and the “maximized” results are always given for the case where the corresponding internal evaluation dimension has been maximized. Except for the relatedness dimension (first group on the left), balancing all four dimensions actually produced better results than maximizing the respective single dimension.

Pairwise statistical permutation tests between the three groups of selection methods (balanced, maximized, minimized), for differences in the mean of the overall judgments, indicate that the balanced selection is statistically significantly better than the minimized selection ( $P = 0.029$ , one-tailed). These statistics confirm that slogans with balanced values on multiple dimensions improve the suitability of slogans over the case where they are minimized.

Existing, expert-written slogans stand out again with a clear margin. They received a median judgment of 3.7 for being good slogans, compared to 3.0 for the balanced computer-generated slogans. Among the different judgments of expert-written slogans, language correctness received the highest scores and had the smallest variation.

Expert-written slogans are considered to be metaphoric with a median score of 3.4, which is closer to neutral than the other judgments. At the same time, the human judgment, where

**Table 8.** Five-number summaries of mean judgments of slogans, grouped by different selections.

|                        |          | Relatedness |     | Language |     | Metaphoricity |     | Catchiness |     | Overall |     |
|------------------------|----------|-------------|-----|----------|-----|---------------|-----|------------|-----|---------|-----|
| Selection method       | <i>n</i> |             |     |          |     |               |     |            |     |         |     |
| Balanced dimensions    |          |             |     |          |     |               |     |            |     |         |     |
| <i>pos(r, l, m, p)</i> | 262      | 3.1         |     | 3.2      |     | 3.0           |     | 3.0        |     | 3.0     |     |
|                        |          | 2.6         | 3.6 | 2.6      | 3.6 | 2.6           | 3.2 | 2.4        | 3.4 | 2.5     | 3.4 |
|                        |          | 1.2         | 4.6 | 1.4      | 4.6 | 1.4           | 4.8 | 1.2        | 4.4 | 1.2     | 4.6 |
| <i>pos(r, l, m)</i>    | 93       | 3.0         |     | 3.0      |     | 3.0           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.4         | 3.6 | 2.6      | 3.6 | 2.6           | 3.4 | 2.4        | 3.4 | 2.4     | 3.4 |
|                        |          | 1.2         | 4.8 | 1.6      | 4.6 | 1.6           | 4.4 | 1.4        | 4.6 | 1.4     | 4.4 |
| <i>pos(r, l, p)</i>    | 111      | 3.0         |     | 3.2      |     | 3.0           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.4         | 3.6 | 2.8      | 3.6 | 2.4           | 3.4 | 2.4        | 3.3 | 2.4     | 3.3 |
|                        |          | 1.4         | 4.4 | 1.6      | 4.4 | 1.6           | 4.2 | 1.2        | 4.4 | 1.4     | 4.4 |
| A maximized dimension  |          |             |     |          |     |               |     |            |     |         |     |
| <i>max(r)</i>          | 100      | 3.2         |     | 3.2      |     | 3.0           |     | 2.8        |     | 2.9     |     |
|                        |          | 2.6         | 3.6 | 2.8      | 3.6 | 2.5           | 3.4 | 2.4        | 3.5 | 2.4     | 3.4 |
|                        |          | 1.2         | 4.4 | 1.6      | 4.8 | 1.2           | 4.0 | 1.4        | 4.4 | 1.2     | 4.4 |
| <i>max(l)</i>          | 105      | 2.8         |     | 3.0      |     | 2.8           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.4         | 3.4 | 2.6      | 3.8 | 2.4           | 3.2 | 2.4        | 3.4 | 2.2     | 3.2 |
|                        |          | 1.2         | 4.4 | 1.8      | 4.8 | 1.6           | 4.2 | 1.4        | 4.4 | 1.4     | 4.8 |
| <i>max(m)</i>          | 88       | 3.0         |     | 3.0      |     | 2.8           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.5         | 3.4 | 2.6      | 3.4 | 2.5           | 3.4 | 2.4        | 3.4 | 2.4     | 3.4 |
|                        |          | 1.4         | 4.6 | 1.6      | 4.6 | 1.6           | 4.4 | 1.4        | 4.4 | 1.4     | 4.4 |
| <i>max(p)</i>          | 96       | 3.0         |     | 3.2      |     | 3.0           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.4         | 3.6 | 2.6      | 3.7 | 2.4           | 3.4 | 2.4        | 3.4 | 2.4     | 3.2 |
|                        |          | 1.2         | 4.2 | 1.4      | 4.4 | 1.4           | 4.8 | 1.4        | 4.4 | 1.4     | 4.6 |
| Minimized dimensions   |          |             |     |          |     |               |     |            |     |         |     |
| <i>min(r, l, m, p)</i> | 104      | 2.6         |     | 3.0      |     | 2.8           |     | 2.8        |     | 2.8     |     |
|                        |          | 2.4         | 3.2 | 2.6      | 3.4 | 2.4           | 3.2 | 2.4        | 3.2 | 2.2     | 3.2 |
|                        |          | 1.4         | 4.8 | 1.2      | 4.6 | 1.2           | 4.4 | 1.4        | 4.4 | 1.2     | 4.4 |
| Expert-written slogans |          |             |     |          |     |               |     |            |     |         |     |
|                        | 100      | 3.8         |     | 4.0      |     | 3.4           |     | 3.7        |     | 3.7     |     |
|                        |          | 3.5         | 4.1 | 3.8      | 4.2 | 3.2           | 3.6 | 3.4        | 3.8 | 3.4     | 3.9 |
|                        |          | 2.3         | 4.7 | 2.7      | 4.6 | 2.4           | 4.2 | 2.6        | 4.5 | 2.4     | 4.6 |

Letters in the *Selection method* column reflect the four evaluation dimensions: relatedness to input, language, metaphoricity, and prosody. *pos(·)* denotes a positive value on all mentioned dimensions, while *min(·)* and *max(·)* indicate that the given dimension is minimized or maximized, respectively. The number of slogans evaluated is expressed as *n*.



**Table 9.** Slogan skeletons used in this paper, in a simplified form without grammatical relations

| Skeleton<br>(without dependency structure and trailing period) | Metaphorical<br>origin | Good<br>slogans | Total of<br>slogans | Success<br>rate |
|--|------------------------|-----------------|---------------------|-----------------|
| ***_NOUN ,_PUNCT ***_NOUN and_CCONJ ***_NOUN                   | Yes                    | 50              | 85                  | 0.59            |
| ***_VERB ***_NOUN ._PUNCT ***_VERB ***_ADV                     | No                     | 18              | 31                  | 0.58            |
| ***_ADJ by_ADP ***_NOUN ._PUNCT ***_NOUN by_ADP ***_NOUN       | Yes                    | 24              | 47                  | 0.51            |
| ***_VERB the_DET ***_ADJ ***_NOUN                              | Yes                    | 10              | 23                  | 0.43            |
| ***_NOUN for_ADP a_DET ***_ADJ ***_NOUN                        | Yes                    | 19              | 46                  | 0.41            |
| ***_VERB <sup>a</sup> ***_NOUN                                 | Yes                    | 7               | 18                  | 0.39            |
| ***_VERB the_DET ***_NOUN to_PART ***_NOUN                     | No                     | 5               | 13                  | 0.38            |
| ***_ADJ than_ADP ***_NOUN                                      | No                     | 8               | 22                  | 0.36            |
| ***_VERB ***_ADJ   | No                     | 7               | 20                  | 0.35            |
| The_DET ***_ADJ ***_NOUN is_VERB ***_NOUN                      | Yes                    | 8               | 23                  | 0.35            |
| ***_PROPN ***_VERB ***_ADJ                                     | No                     | 2               | 6                   | 0.33            |
| ***_NOUN ***_NOUN ._PUNCT ***_VERB ***_NOUN                    | No                     | 9               | 27                  | 0.33            |
| The_DET ***_NOUN of_ADP ***_NOUN                               | Yes                    | 7               | 21                  | 0.33            |
| The_DET ***_ADJ ***_NOUN on_ADP ***_NOUN                       | Yes                    | 13              | 40                  | 0.33            |
| ***_NOUN never_ADV ***_VERB out_ADP of_ADP ***_NOUN            | Yes                    | 11              | 38                  | 0.29            |
| ***_VERB your_ADJ ***_NOUN do_VERB the_DET ***_NOUN            | Yes                    | 13              | 48                  | 0.27            |
| You_PRON ca_VERB ***_ADV ***_VERB the_DET ***_ADJ ***_NOUN     | No                     | 8               | 31                  | 0.26            |
| ***_VERB ***_NOUN ***_NOUN                                     | No                     | 6               | 24                  | 0.25            |
| ***_PROPN ***_ADV  | No                     | 4               | 18                  | 0.22            |
| ***_VERB ***_NOUN the_DET ***_NOUN over_ADV                    | No                     | 3               | 16                  | 0.19            |
| ***_NOUN ***_VERB and_CCONJ ***_VERB and_CCONJ ***_VERB        | No                     | 1               | 6                   | 0.17            |
| It_PRON ***_VERB ***_NOUN                                      | No                     | 3               | 19                  | 0.16            |
| Between_ADP ***_NOUN and_CCONJ ***_NOUN ***_VERB ***_NOUN      | Yes                    | 2               | 13                  | 0.15            |
| ***_VERB <sup>b</sup> ***_NOUN                                 | Yes                    | 2               | 14                  | 0.14            |
| I_PRON ***_VERB ***_VERB it_PRON                               | No                     | 1               | 12                  | 0.08            |
| ***_NOUN ._PUNCT It_PRON ***_VERB a_DET ***_NOUN ***_NOUN      | Yes                    | 1               | 23                  | 0.04            |

Skeletons are ordered by their success rates, that is, the ratio of suitable results to produced results.

<sup>a</sup>In base form.

<sup>b</sup>In present participle form.

computer-made slogans are closest to expert-made ones is metaphoricity. This is natural: on the one hand, metaphoricity is not a strong requirement for successful (expert-written) slogans; on the other hand, the method of this paper encourages the use of metaphors in slogans.

### 5.6. Differences between skeletons

Finally, we consider performance differences between skeletons. Table 9 shows all skeletons used in these experiments, along with the numbers of total and successful slogans generated from them (as per mean human judgment greater than 3). Best skeletons produced successful slogans for more than half of the time, whilst for the worst ones, less than one slogan in ten was successful. The absolute numbers of produced and successful slogans also vary, suggesting that some skeletons are easier to instantiate than others.

The method described in this paper aims to produce metaphoric slogans by construction. Are skeletons extracted from existing metaphoric slogans better at producing metaphorical slogans?

One half of the 26 skeletons originate from metaphorical slogans (cf. Table 9); 38% of slogans generated from them were considered metaphorical, compared to 31% for slogans generated from the other skeletons. In total, 35% of all generated slogans were considered to be metaphorical.

These results indicate that generating slogans using skeletons extracted from metaphorical slogans has a higher potential to produce metaphorical slogans as well. On the other hand, the proposed method appears to be capable of generating metaphorical slogans also from nonmetaphorical skeletons, even if the success rate in this respect is modest, around one third.

## 6. Discussion

*Metaphor generation.* To the best of our knowledge, our metaphor construction method is the first one based on a metaphor interpretation model. The experimental results indicate that this is beneficial: metaphorical vehicles that are more likely to have the desired interpretation, in the context of the given tenor, outperformed vehicles selected solely based on their strong association to the target property.

Nonetheless, the metaphor interpretation model only gives partial information on how a metaphor is comprehended. For instance, two examples of apt vehicle candidates produced for expressing that a *computer* is *creative* are *poet* and *music*. The interpretations can be quite different: the former suggests that a computer can produce creative artifacts, while the latter suggests that the computer is a creative artifact itself. This question is partially related to the ambiguity of the word *creative*.

The experiments show that personal vehicles (such as *poet* above) produced on average better metaphors than general nouns (such as *music*), and the effect was relatively strong (cf. Table 5). What kind of vehicles are more effective varies across slogans and the role of the vehicles in them. However, personal vehicles probably are more likely to assign human properties to the tenor, and possibly, this tends to make the metaphors better. Further analysis is required to assess the impacts of each type, given that we have utilized two different resources which could have affected the results.

While salience imbalance and similarities between the vehicle and tenor are approximated through the metaphor interpretation model, additional criteria could be considered to further assess the aptness of generated metaphors. Examples of such criteria are the ontological distance between the concepts, concreteness of the vehicle, and the novelty of the metaphor.

*Skeleton-based slogan production.* In the experiments, the number of good slogans, that is, slogans with a mean score greater than three, ranged from 1 to 13 per input. We consider this to be a strong result: each input resulted in at least one good slogan. This was despite artificial limitations in our experimental setting; in particular, we used only two slogan skeletons for each input, out of our pool of 26 skeletons. This limitation was introduced for ease of experimentation only, and in real use of the method in supporting ideation of slogans, a larger set of skeletons obviously should be used. This would increase not only the number of better slogans, but also the variety of slogans produced.

Our 26 skeletons varied a lot in their productivity and success rates (Table 9). The fraction of successful slogans among those generated from a single skeleton varied from 59% to 4%. It is not obvious where these differences come from. While simple expressions are easier to generate, they are not necessarily better slogans. According to our results (Table 9), the length or complexity of the skeleton is not directly reflected in its success rate. This topic, among others, deserves further study and should be considered in practical use of the method.

Regardless of the success rate of generated slogans, some skeletons are harder to instantiate than others. The slogan generation method ensures that the grammatical relations encoded in skeletons are obeyed (see Figure 1 for an example). Sometimes, however, the method is not able to instantiate a skeleton. Obviously, the grammatical complexity of a skeleton constraints the number of ways it can be filled in. The method may run into a dead end also because of its preference for related words. Recall that when a placeholder  $i$  is being filled in a skeleton, the method identifies the set  $\mathcal{G}_i$  of words grammatically consistent with the words already in other placeholders of the skeleton, and its further restriction  $\mathcal{R}_i \subset \mathcal{G}_i$  to words related to the target concept and property given as input. The method resorts to grammatical words in  $\mathcal{G}_i$  if  $\mathcal{R}_i$  is empty, and problems materialize when  $\mathcal{G}_i$  is also empty. It would be possible to remedy the dead-end problem without giving up the grammatical constraints: increasing the sizes of the related spaces would provide more (related) alternatives for fillers earlier in the process, potentially leading to more (grammatical) alternatives also later on. The downside of this would be decreased relatedness of slogans to the target concept and property. This option is worth exploring further, however, since relatedness can be—and already is—measured and optimized as one of the internal evaluation dimensions.

Further variation in skeletons and slogans could potentially be obtained by generating new skeletons automatically. One could try to linguistically analyze both slogan and non-slogan expressions manually or by machine learning to highlight their differences (cf. Yamane and Hagiwara 2015; Repar *et al.* 2018; Alnajjar 2019), and then generate novel slogan-like skeletons. We leave this for future research.

*Internal evaluation dimensions and human judgments.* The empirical tests indicated statistically significant associations between the internal evaluation dimensions and the corresponding human judgments for relatedness, metaphoricity, and prosody/catchiness. The result suggests that these internal evaluation dimensions could be given a larger role in the design of the method. For instance, a wider selection of slogans could potentially be obtained by removing the strict coherence requirement that all words in a slogan must be related to each other (cf. Section 4.4). Instead, the method could rely more on the existing evaluation dimensions, and a measure of internal coherence could be added as a new one.

The correlation between internal evaluation and human judgment was not significant for language. This reflects the design of the method: the search space has a lot of variation in terms of relatedness, metaphoricity, and prosody, while the language is strongly bounded by the grammatical constraints of the skeletons. In addition, the internal evaluation dimension of language combines language correctness and surprise, while human judges were only asked about language correctness.

In human judgments of the four aspects of generated slogans, language correctness received better scores than relatedness, catchiness, and metaphoricity (Table 7). This speaks in favor of the grammatical constraints and their maintenance throughout the method, even if the internal language dimension was not able to reliably measure the remaining variance in quality, and even though some skeletons were not so productive due to the constraints. At the same time, more creative slogans could potentially be produced by dropping strict grammar constraints. This could, however, result in too many poor expressions, and automated assessment of their quality would be difficult.

The relatively low performance of generated slogans with respect to metaphoricity (Table 7) is somewhat surprising, given that the method is specially constructed to use metaphor. However,

by design, the method does not enforce all slogans to be metaphoric. Rather, they are encouraged to be metaphoric by primarily using words in  $\mathcal{R}_i$  related to the concept or the vehicle, and by the internal metaphoricity evaluation dimension. As mentioned above, the correlation between the internal evaluation dimension and the human judgment of metaphoricity was statistically significant, allowing for optimization of metaphoricity in the results.

Looking at correlations between human judgments of different aspects of slogans (Figure 5), we observed that correlations tended to be high but that correlation between metaphoricity and relatedness was relatively low (0.37). This is probably explained by the introduction of a metaphoric vehicle and words associated to it, which decreases associations to the input concept. (Nevertheless, metaphoricity has a strong positive correlation with catchiness and overall suitability of slogans.)

Despite the abovementioned statistically significant relation between internal evaluation and relatedness, metaphoricity, and prosody/catchiness, maximizing just one internal dimension seems to only have some correspondence to the respective human judgment (Table 8). This confirms the broader observation that better slogans are obtained by a balanced mix of several internal dimensions than by a single one. High correlations between the human judgments (Figure 5) suggest that those aspects are intertwined and cannot be easily optimized in isolation.

Finally, while we have observed statistically significant associations between the degree of metaphoricity as measured by the internal dimension and by human judgment, there is no guarantee that generated slogans convey the intended metaphor and the intended property. It would be interesting to analyze the human interpretations of metaphors, both in their nominal form (i.e., purely as metaphors) and in the produced slogans. Such evaluation probably should involve open questions asking the judges to give their interpretations. Obtaining answers of sufficiently high quality could be difficult in crowdsourcing, and quantitative assessment of the answers would be difficult, too.

*Resources and parameters.* The method proposed in this paper makes use of multiple linguistic resources and tools, and limitations in their scopes and functionalities can have an impact on the slogans generated. The resources include well-known corpora (e.g., *ukWaC*) and tools (e.g., *NLTK* and *spaCy*), but also the more novel metaphor interpretation model *Meta4meaning* by Xiao *et al.* (2016). Metaphor interpretation is a difficult and ambiguous task, and misinterpretations by *Meta4meaning* are not unlikely, potentially resulting in metaphors conveying a meaning different from the intended one. This issue is also related to polysemy, which is not directly dealt with by our methods. Additionally, given that slogans are short and not even full sentences, NLP tools might fail in parsing them. Such failures result in building skeletons with incorrect grammatical relations, eventually affecting the generated expressions.

The slogan generation method takes multiple parameters that could be tuned to achieve better results. For instance, computation of the semantic model  $\omega$  alone (cf. Section 4.2) takes parameters such as window width and frequency limits; the genetic algorithm likewise takes many parameters. More central to the slogan generation method are issues like the number of related words to consider when filling in skeletons (cf. discussion above). Reducing the number would likely result in generating fewer yet better slogans, while an increase would produce a larger variety of slogans including ones that are less related to the given concept and property. As relatedness to the advertised product and the desired property is important for slogans, the former approach seems more promising, especially if a larger selection of skeletons is used to ensure that a variety of slogans is produced.

*Effects of Randomness.* In this paper, there are two major uses of randomness: in generation of metaphors and slogans, and in empirical evaluation of the generation system.

Starting with empirical evaluation, selecting a random sample from the output produced by the system is a common practice in evaluation of generative methods. In our evaluation, we have

used random artifacts of several types, for example, strongly related, related, and random vehicles, as well as slogans with balanced, maximized, and minimized evaluation dimensions, in order to shed light on how the method works and what affects the quality of the output. Regarding generation of metaphors and slogans, we have two notes. First, randomness mostly takes place within stochastic search/optimization algorithms: during its operation, the method makes random decisions, but it also evaluates the decisions and either pursues the most promising ones, or selects the better ones for the next phases. Overall, the operation thus is not arbitrary while randomness is used as part of the method. Second, in most cases random selection is informed, not blind. For instance, the method carries out random selection among the top vehicles, or among the most strongly associated words, in order to provide variation and to avoid relying too heavily on computational estimates. Because of this stochasticity, we have evaluated a large number of artifacts from different aspects, to reduce random effects in the results.

Additionally, there is one major random choice in the paper: selection of which skeletons to use. As discussed earlier, we only use two random skeletons for each input, in order to make the empirical tests of this paper feasible.

*Crowdsourcing and evaluation.* In our evaluations, we have used a crowdsourcing platform to judge metaphors and slogans. We chose to obtain opinions of ordinary people, rather than advertising experts, because they are much easier to reach. A hard-to-mitigate risk of crowdsourcing subjective tasks that do not have unique answers is scammers, that is, users who abuse the system by answering tasks very fast, possibly just randomly, in order to maximize their income. Given that scammers add noise to the data, the signals that were detected statistically despite the noise are likely to be reliable. However, some associations may have remained undiscovered due to the noise.

Another problem with crowdsourcing was that we could not assume that the judges know and understand linguistic concepts such as metaphor, semantic relatedness, and prosody. We aimed to craft the questions in a manner that would be simple to understand and answer, but regardless of our best efforts, it is infeasible for us to verify that the judges have actually understood the task fully and answered accordingly.

The purpose of the proposed method is to act as an ideation tool for professionals when constructing slogans. We did not evaluate the method in this use case, but it would be relevant to assess if the method can actually inspire professionals. This would involve recruitment of professionals willing to test the method, further development of the method to a user-friendly tool, and design of the experimental setup. In sustained use, the tool could additionally monitor its use by the professionals, slogans selected/saved, adjustments to parameters, etc., and then estimate the relationships between parameters, internal evaluation dimensions, and the satisfaction of slogans by the users.

Given the difficulty of assessing the method with professionals, a more practical evaluation could compare generated slogans to those written by amateurs. Additional task-related relevance could be obtained by using both generated and amateur-written slogans for further ideation and development (by amateurs), and seeing how different initial slogans fare in mutual comparison.

*Creativity.* Generation of slogans is a creative task involving “production of a novel and appropriate response, product, or solution to an open-ended task” (Amabile 2012). It would be interesting to assess the creativity of the method, or the creativity of the slogans and metaphors produced. The field of computational creativity (Colton and Wiggins 2012; Xiao *et al.* 2019) offers conceptual tools for this. A full discussion is outside the scope of this paper, but Jordanous (2012) describes a procedure consisting of defining what creativity means in the application at hand and then deriving evaluation metrics. To instantiate the evaluation methodology to slogan generation, the example by Alnajjar and Hämmäläinen (2018) could be followed, as it evaluates a related creative task.



## 7. Conclusions

In this paper, we have introduced a method for generating metaphorical slogans computationally, given a concept to produce a slogan for, and a property to be associated to the concept. As a subcomponent of the approach, we have also proposed a novel method for generating metaphors.

The slogan generation method uses skeletons, that is, templates with empty placeholders and grammatical constraints between them. We have described how skeletons can be extracted automatically from existing slogans, how possible (metaphorical) filler words are identified, and how the resulting slogan candidates can be assessed using four internal evaluation dimensions. We have used a genetic algorithm to construct slogans with a multi-objective fitness function based on the evaluation dimensions.

The metaphor generation method uses a metaphor interpretation model to identify metaphorical vehicles that are likely to result in the intended interpretation. To the best of our knowledge, this is the first metaphor generation method based on an interpretation model rather than just generation heuristics.

We have evaluated the proposed method and its various components using crowdsourcing. Our empirical findings can be summarized as follows:

- The method produced at least 1 good slogan for each input and up to 13 for some. Significant increase can be expected when using more skeletons instead of the two (out of 26) used per input in our experiments.
- Catchiness, relatedness to the target concept, language correctness, and metaphoricity correlate with the overall quality of slogans ( $r = 0.86, 0.74, 0.68$ , and  $0.58$ , respectively), based on the evaluation on expert-made slogans. Further, the internal evaluation measures defined in this paper for relatedness, metaphoricity, and prosody/catchiness are related to the corresponding human judgments to a statistically significant degree ( $P = 10^{-6}$ ,  $0.0033$ , and  $0.046$ , respectively). These results imply that it is possible to computationally measure—and thus, optimize—three criteria that contribute to the overall quality of slogans.
- Best slogans are obtained, on average, when the four internal evaluation dimensions are balanced. Maximizing just one of them tends to produce inferior results, often also for the maximized aspect.
- The productivity and success rate of individual skeletons varies considerably. The best skeletons produced an order of magnitude more slogans than poorer ones, and they produced good slogans for more than half of the time. By using the better skeletons only, the average overall quality of generated slogans can be increased considerably.
- Regarding metaphor generation on its own, using the metaphor interpretation model gives, on average, better metaphors than a corresponding method without it. Further, personal vehicles tend to produce better metaphors than vehicles of the general class.

This work has taken steps toward automated generation of metaphorical slogans, and toward generation of metaphors based on their interpretations. We hope that the methods described in this paper and our empirical observations earlier in this section help others build even better metaphor and slogan generation systems.

In future work, we will adapt the ideas presented in this paper to generation of other creative expressions. We are especially interested in producing short, catchy texts in a given textual context, such as creating attractive headlines (Gatti *et al.* 2016; Alnajjar, Leppänen, and Toivonen 2019) for automatically generated news texts (cf. Bouayad-Agha *et al.* 2012; Leppänen *et al.* 2017). Slogans tend to have no textual context, making their generation a more isolated task. Having a context adds complexity to the task, but also provides clues to completing the task. We also plan to expand



the current setting with exactly one concept and one property to handle cases of multiple concepts and details (e.g., in the news domain, comparing election results of two parties in a given city).

We are also interested in multilingual settings. While the current work only considers English, the key ideas hold for many other languages for which similar tools and resources are available.

Finally, another future direction could be altering existing texts to include some metaphoricity, extending current word-substitution-based methods for generation of creative language (Toivanen *et al.* 2012; Valitutti *et al.* 2016). After identifying a metaphorical reference topic for a given text, the method could be adjusted to replace verbs and adjectives in the text with content words from the space related to the reference topic, while maximizing the metaphoricity dimension.

**Acknowledgments.** We would like to thank the anonymous reviewers for their helpful comments.

**Financial support.** This work has been supported by the Academy of Finland under grant 276897 (CLiC) and by the European Union's Horizon 2020 programme under grant 825153 (Embeddia).

## References

- Alnajjar K. (2018). The 12 million most frequent English grammatical relations and their frequencies. <https://doi.org/10.5281/zenodo.1255800>.
- Alnajjar K. (2019). *Computational Analysis and Generation of Slogans*. Master's Thesis, Helsingin yliopisto, Helsinki, Finland.
- Alnajjar K., Hadaytullah H. and Toivonen H. (2018). "Talent, Skill and Support." A method for automatic creation of slogans. In *Proceedings of the 9th International Conference on Computational Creativity (ICCC 2018)*, Salamanca, Spain. Association for Computational Creativity, pp. 88–95.
- Alnajjar K. and Hämäläinen M. (2018). A master-apprentice approach to automatic creation of culturally satirical movie titles. In *Proceedings of the 11th International Conference on Natural Language Generation*, Tilburg, The Netherlands. Association for Computational Linguistics, pp. 274–283.
- 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 8th International Conference on Computational Creativity (ICCC 2017)*, Georgia, Atlanta, USA. Georgia Institute of Technology, pp. 25–32.
- Alnajjar K., Leppänen L. and Toivonen H. (2019). No time like the present: methods for generating colourful and factual multilingual news headlines. In *The 10th International Conference on Computational Creativity*, Charlotte, North Carolina, USA. Association for Computational Creativity, pp. 258–265.
- Amabile T. (2012). *Componential Theory of Creativity*. Working Paper No. 12–096, Harvard Business School.
- Baroni M., Bernardini S., Ferraresi A. and Zanchetta E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation* 43(3), 209–226.
- Bird S., Klein E. and Loper E. (2009). *Natural Language Processing with Python*, 1st Edn. O'Reilly Media, Inc.
- Bouayad-Agha N., Casamayor G., Mille S. and Wanner L. (2012). Perspective-oriented generation of football match summaries: old tasks, new challenges. *ACM Transactions on Speech and Language Processing* 9(2), 1–31.
- Burgers C., Konijn E.A., Steen G.J. and Iepma 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.
- Colton S. and Wiggins G.A. (2012). Computational creativity: the final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence, ECAI'12*, Amsterdam, The Netherlands. IOS Press, pp. 21–26.
- Dahl G. (2011). *Advertising for Dummies*. Hoboken, NJ: John Wiley & Sons.
- De Smedt T. and Daelemans W. (2012). Pattern for Python. *Journal of Machine Learning Research* 13, 2063–2067.
- 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*, Vol. 2. Berlin: Mouton de Gruyter, pp. 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.
- Fuertes-Olivera P.A., Velasco-Sacristán M., Arribas-Baño A. and Samaniego-Fernández E. (2001). Persuasion and advertising English: metadiscourse in slogans and headlines. *Journal of Pragmatics* 33(8), 1291–1307.

- Galván P., Francisco V., Hervás R., Méndez 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.
- Gatti L., Özbal G., Guerini M., Stock O. and Strapparava C. (2015). Slogans are not forever: adapting linguistic expressions to the news. In *Proceedings of the 24th International Conference on Artificial Intelligence, Stanford, California, USA*. AAAI Press, pp. 2452–2458.
- Gatti L., Özbal G., Guerini M., Stock O. and Strapparava C. (2016). Automatic creation of flexible catchy headlines. In “Natural Language Processing meets Journalism”—IJCAI 2016 Workshop, New York City, pp. 25–29.
- 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 6th International Conference on Computational Creativity (ICCC 2015)*, Park City, Utah, USA. Brigham Young University, pp. 71–77.
- Honnibal M. and Montani I. (2017). spaCy 2: natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Iwama K. and Kano Y. (2018). Japanese advertising slogan generator using case frame and word vector. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands*. Association for Computational Linguistics, pp. 197–198.
- Jordanous A. (2012). A standardised procedure for evaluating creative systems: computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3), 246–279.
- Katz A.N. (1989). On choosing the vehicles of metaphors: referential concreteness, semantic distances, and individual differences. *Journal of Memory and Language* 28(4), 486–499.
- Koestler A. (1964). *The Act of Creation*. London: London Hutchinson.
- Kohli C., Suri R. and Thakor M. (2002). Creating effective logos: insights from theory and practice. *Business Horizons* 45(3), 58–64.
- Lenzo K. (1998). The CMU pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
- Leppänen L., Munezero M., Granroth-Wilding M. and Toivonen H. (2017). Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation, Santiago de Compostela, Spain*. Association for Computational Linguistics, pp. 188–197.
- Marcus M.P., Santorini B. and Marcinkiewicz M.A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19(2), 313–330.
- Mathur L.K. and Mathur I. (1995). The effect of advertising slogan changes on the market values of firms. *Journal of Advertising Research* 35(1), 59–65.
- 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. (2016). An analysis of rhetorical figures and other linguistic devices in corporation brand slogans. *Journal of Marketing Communications* 22(5), 474–493.
- Ortony A. (1993). *The Role of Similarity in Similes and Metaphors*, 2nd Edn. Cambridge: Cambridge University Press, pp. 342–356.
- Ortony A., Vondruska R.J., Foss M.A. and Jones L.E. (1985). Salience, similes, and the asymmetry of similarity. *Journal of Memory and Language* 24(5), 569–594.
- Özbal G., Pighin D. and Strapparava C. (2013). BRAINSUP: brainstorming support for creative sentence generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria*. Association for Computational Linguistics, pp. 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.
- Reinsch Jr. N.L. (1971). An investigation of the effects of the metaphor and simile in persuasive discourse. *Speech Monographs* 38(2), 142–145.
- Repar A., Martinc M., Žnidaršič M. and Pollak S. (2018). BISLON: BISociative SLOgaN generation based on stylistic literary devices. In *Proceedings of the Ninth International Conference on Computational Creativity, Salamanca, Spain*. Association for Computational Creativity (ACC), pp. 248–255.
- Richards I.A. (1936). *The Philosophy of Rhetoric*. London: Oxford University Press.
- Strapparava C., Valitutti A. and Stock O. (2007). Automatizing two creative functions for advertising. In Cardoso A. and Wiggins G. (eds), *Proceedings of the 4th International Joint Workshop on Computational Creativity, London, UK*. London: Goldsmiths, University of London, pp. 99–108.
- Toivonen J.M., Toivonen H., Valitutti A. and Gross O. (2012). Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC 2012)*, Dublin, Ireland, pp. 211–215.
- Tom G. and Eves A. (1999). The use of rhetorical devices in advertising. *Journal of Advertising Research* 39(4), 39–43.
- Tomašič P., Papa G. and Žnidaršič M. (2015). Using a genetic algorithm to produce slogans. *Informatica* 39(2), 125.

- Tomašič P., Žnidaršič M. and Papa G.** (2014). Implementation of a slogan generator. In *Proceedings of the 5th International Conference on Computational Creativity (ICCC 2014)*, Ljubljana, Slovenia. Josef Stefan Institute, 340–343.
- Tourangeau R. and Sternberg R.J.** (1981). Aptness in metaphor. *Cognitive Psychology* 13(1), 27–55.
- 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*, Edinburgh, United Kingdom. Association for Computational Linguistics, pp. 680–690.
- Valitutti A., Doucet A., Toivanen J.M. and Toivonen H.** (2016). Computational generation and dissection of lexical replacement humor. *Natural Language Engineering* 22(5), 727–749.
- Veale T. and Li G.** (2012). Specifying viewpoint and information need with affective metaphors: a system demonstration of the metaphor magnet web app/service. In *Proceedings of the ACL 2012 System Demonstrations, ACL '12*, Jeju Island, Korea. Association for Computational Linguistics, pp. 7–12.
- 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, Sofia, Bulgaria*. Association for Computational Linguistics, pp. 660–670.
- 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 7th International Conference on Computational Creativity (ICCC 2016)*, Paris, France. Sony CSL.
- Xiao P. and Blat J.** (2013). Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC 2013)*, Sydney, Australia. The University of Sydney, pp. 8–15.
- Xiao P., Toivonen H., Gross O., Cardoso A., Correia J., Machado P., Martins P., Oliveira H.G., Sharma R., Pinto A.M., Díaz A., Francisco V., Gervás P., Hervás R., León C., Forth J., Purver M., Wiggins G.A., Miljković D., Podpečan V., Pollak S., Kralj J., Žnidaršič M., Bohanec M., Lavrač N., Urbančič T., Velde F.V.D. and Battersby S.** (2019). Conceptual representations for computational concept creation. *ACM Computing Surveys* 52(1), 9:1–9:33.
- Yamane H. and Hagiwara M.** (2015). Tag line generating system using knowledge extracted from statistical analyses. *AI & SOCIETY* 30(1), 57–67.
- Žnidaršič M., Tomašič P. and Papa G.** (2015). Case-based slogan production. In Kendall-Morwick J. (ed), *Proceedings of the ICCBR 2015 Workshops, Frankfurt, Germany*. CEUR, pp. 123–130.

**Cite this article:** Alnajjar K and Toivonen H. Computational generation of slogans. *Natural Language Engineering* <https://doi.org/10.1017/S1351324920000236>

Downloaded from <https://www.cambridge.org/core>. Helsinki University Library, on 04 Jun 2020 at 08:37:11, subject to the Cambridge Core terms of use, available at <https://www.cambridge.org/core/terms>. <https://doi.org/10.1017/S1351324920000236>

# Appendix C: When a Computer Cracks a Joke: Automated Generation of Humorous Headlines

## When a Computer Cracks a Joke: Automated Generation of Humorous Headlines

**Khalid Alnajjar**

Faculty of Arts  
University of Helsinki  
khalid.alnajjar@helsinki.fi

**Mika Hämäläinen**

Faculty of Arts  
University of Helsinki  
mika.hamalainen@helsinki.fi

### Abstract

Automated news generation has become a major interest for new agencies in the past. Oftentimes headlines for such automatically generated news articles are unimaginative as they have been generated with ready-made templates. We present a computationally creative approach for headline generation that can generate humorous versions of existing headlines. We evaluate our system with human judges and compare the results to human authored humorous titles. The headlines produced by the system are considered funny 36% of the time by human evaluators.

### Introduction

Humor, while showcased by a wide spectrum of species in the animal kingdom, has a subcategory that is exclusive to the human being. Verbal humor can only exist in the presence of language, and its generation by computational means is far from trivial.

Humor is effectively a perceiver dependent phenomenon. Nothing can be inherently funny, but humor is perceived and appraised by the mind of a human perceiving it. And ultimately, the perceived humor, if accepted as such by the listener, elicits an emotional response accompanied with a vocal response of laughter that originates from our ancestors, the species before homo sapiens (cf. Ross, Owren, and Zimmermann 2010).

Our paper focuses on generating humor in news headlines. This NLG task is not of the traditional sort, where conveying factual information is the uttermost goal of the system, but rather the affective content of the message is taken into the primary focus of the study.

Automated news generation is a flourishing field with new research being published in a timely manner. This is reflected by the number of recent publications on the topic (Nesterenko 2016; Yao et al. 2017). Quite often, however, as the generated news has to cater for the purpose of communicating facts the question of creativity is set aside. In such a context, there is a trade-off between creativity and communicativity (see Hämäläinen and Honkela 2019).

Whereas creative headline generation is not a new domain to computational creativity, with quite some existing publications on the topic (Lynch 2015; Gatti et al. 2015;

Alnajjar, Leppänen, and Toivonen 2019), we aim to intertwine headlines, creativity and humor by proposing a novel method for humor generation that is reasoned by the existing theories on humor.

Our approach alters a word in an existing headline for a humorous effect. We evaluate the method proposed in this paper quantitatively with human judges. We take the different constituents of humor in consideration in the evaluation to uncover the relation of each feature to the humor produced by our system.

### Related Work

Humor has received some interest in the past for more than a decade (Ritchie 2005; Hong and Ong 2009; Valitutti et al. 2013; Costa, Oliveira, and Pinto 2015). We dedicate the remaining of this section to describing some of the most recent work conducted on the topic.

Pun generation with a neural model language model is one of the most recent efforts on humor generation (Yu, Tan, and Wan 2018). Their approach consists of training a conditional language model using a beam search to find sentences that can support two polysemous meanings for a given word. In addition they train a model to highlight the different meanings of the word in the sentence. Unfortunately, they evaluate their system on human evaluators based on three quantitative metrics: fluency, accuracy and readability, none of which tells anything about how funny or apt the puns were.

Alnajjar and Hämäläinen (2018) present a genetic algorithm approach for generating humorous and satirical movie titles out of existing ones. Their method works on a word level replacement and aims for low semantic similarity of the replacement word with the original word to maximize surprise and high similarity with Saudi Arabia to maximize coherence. They consider pun as one of the fitness functions of the genetic algorithm, but the output is not strictly limited to puns. On top the genetic algorithm, they train an RNN model that learns from the genetic algorithm and real people.

Surprise is also one of the key aspects of a recent pun generator (He, Peng, and Liang 2019). They model surprise as conditional probabilities. They introduce a local surprise model to assess the surprise in the immediate context of the pun word and a global surprise to assess the surprise in the

context of the whole text. Their approach retrieves text from a corpus based on an original word - pun word pair. They do a word replacement for local surprise and insert a topic word for global surprise.

An approach building on humor theories is that of Winters, Nys, and De Schreye (2019). The theories are used in feature engineering. They learn templates and metrical schemata from jokes rated by people with a star rating. They embrace more traditional machine learning techniques over neural networks, which has the advantage of a greater interpretability of the models.

Humor has also been tried to recognize automatically in the past. One of such attempts is focuses on extracting humor anchors, i.e. words that can make text humorous, automatically (Yang et al. 2015). A similar humor anchor based approach is also embraced by Cattle and Ma (2018). Both of the approaches rely on feature engineering basing on humor theories. Recently LSTM models have been used for the task of humor detection with a different rates of success (Cai, Li, and Wan 2018; Sane et al. 2019; Zou and Lu 2019).

## Humor

Humor is an inherent part of being a human and as such it has provoked the interest of many researchers in the past to formulate a definition for it (see (Krikmann 2006)). Koestler (1964) sees humor as a part of creativity together with discovery and art. In his view, what is characteristic to humor in comparison to the other two constituents of creativity, is that its emotional mood is aggressive in its nature. He calls bisociation in humor the collision of two frames of reference in a comic way.

Raskin (1985) presents a theory that is not too far away from the previously described one in the sense that in order for text to be humorous, it has to be compatible with two different scripts. The different scripts have to be somehow in opposition, for example in the sense that one script is a real situation and the other is not real.

In Attardo and Raskin (1991) humor is seen to consist of six hierarchical knowledge resources: language, narrative strategy, target, situation, logical mechanism and script opposition. As in the previous theories, the incongruity of two possible interpretations is seen as an important aspect for humor. An interesting notion that we will take into a closer examination is that of target. According to the authors it is not uncommon for a joke to have a target, such as an important political person or an ethnic group, to be made fun of.

Two requirements have been suggested in the past as components of humor in jokes: surprise and coherence (see (Brownell et al. 1983)). A joke will then consist of a surprising element that will need to be coherent in the context of the joke. This is similar to having two incongruous scripts being simultaneously possible.

Veale (2004) points out that the theories of Raskin (1985) and Attardo and Raskin (1991) entail people to be forced into resolution of humor. He argues that humor should not be seen as resolution of incompatible scripts, but rather as

a collaboration, where the listener willingly accepts the humorous interpretation of the joke. Moreover, he argues that while incongruity contributes to humor, it does not alone constitute it.

## Generating Humorous Headlines

In their work, Hossain, Krumm, and Gamon (2019) identified several ways people altered news headlines to be humorous. In our method, we aim to model the following ones of their findings: the replacement forms a meaningful n-gram, the replacements are semantically distant, the replacement makes a strong connection with the entity of the headline and belittles an entity or a noun and the replacement creates incongruity. We see the n-gram finding in a broader way of the replacement being compatible with the the existing script (context). The semantic distance is seen as an index of surprise, and the connection between the entity is assimilated with the target of the joke.

The findings we are not focusing on in this paper are that the replacements are sarcastic, suppress tension or have a setup and punchline. The first two are left out as assessing them computationally is a task worth of a paper on their own right, and the third one is left out as it focuses on a particular kind of humor. However, the punchline structure might emerge from the other features being modelled although not explicitly taken into consideration.

In addition to the findings described above, we take the concreteness of the replacement word into account. The reason for this that concrete words are more likely to provoke mental images (see (Burroway 2007)). In fact, we could see this in the humorous training dataset by Hossain, Krumm, and Gamon (2019), where 90% of the most humorous replacement words were concrete as opposed to only 75% of the least humorous replacement words being concrete.

For the above experiment and the rest of the paper, we use the lexicon of 40k common English words that has a concreteness score from 1 to 5 assigned (Brysbaert, Warriner, and Kuperman 2014). If the score assigned with the word is greater or equal to 3, we consider it concrete. The concreteness is evaluated by lemmatizing the word with spaCy (Honnibal and Montani 2017) if it does not exist in the lexicon.

## Modelling Humor

Our system operates by taking an existing headline from the corpus of altered headlines (Hossain, Krumm, and Gamon 2019). This corpus has been syntactically parsed by us by using spaCy (Honnibal and Montani 2017), and it has been tagged for the words that should be replaced by its original authors. For a selected headline, our system tries to find suitable humorous replacement words.

We assess the different potential humorous replacements in terms of multiple parameters, which are prosody, concreteness, semantic similarity of the replacement to the original word and the semantic relatedness of the replacement to negative words describing the target. In this section, we explain how the individual parameters are modelled. An overall view of our method is depicted in Figure 1.

For prosody, we look at the sound similarity between the original word and the replacement. We assess this in



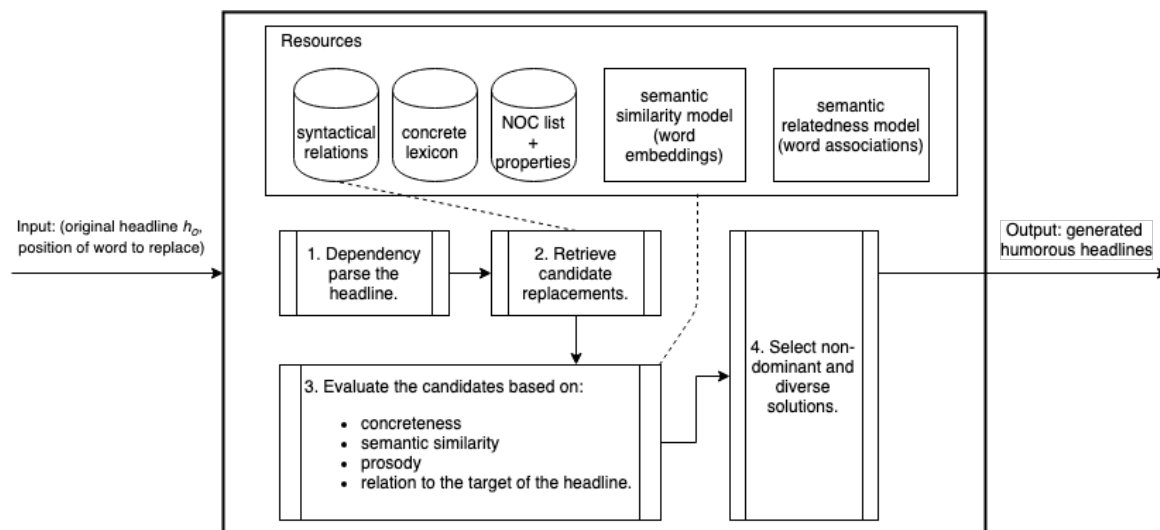


Figure 1: A diagram visualizing the process of humor generation.

terms of full rhymes, assonance, consonance and alliteration. These are implemented with rules. As the written form of English is notoriously deviant from the phonation, we use eSpeak-ng<sup>1</sup> to produce IPA transcription for the words the prosody of which is being assessed.

For concreteness we use the values provided in Brysbaert, Warriner, and Kuperman (2014) to score the concreteness of the replacement word. And for semantic similarity we use the pretrained word embeddings from Bojanowski et al. (2017). We use the semantic similarity to assess surprise, in other words, we want to minimize the similarity of the replacement word to the original.

To measure how a new replacement connects to the word selected to be the target of the joke in the headline, a target must first be found. We consider recognized entities in the headline as the potential targets. In case no entities were recognized, we use the subjects in the headline. If neither of them existed, nouns in the headline are treated as target. Out of the list of targets, a random target  $t$  is picked to focus on. For this target  $t$ , we retrieve words that are related to it to act as descriptive words revealing potential attributes to make fun of. We employ two resources to obtain such knowledge regarding the selected target:

1. The Non-Official Characterization (NOC) list (Veale 2016) which contains information about more than 1000 well-known characters (e.g. *Donald Trump* and *Kim Jung-un*) and their expanded stereotypical properties supplied by (Alnajjar et al. 2017) (e.g. *Donald Trump*: [wealthy, successful, greedy, aggressive, ... etc]).
2. A semantic relatedness model built from word associations collected from a web text corpus ukWac<sup>2</sup>, following the approach described in Meta4meaning (Xiao et al.

<sup>1</sup>UK English voice, <https://github.com/espeak-ng/espeak-ng>

<sup>2</sup><https://wacky.sslmit.unibo.it/doku.php?id=corpora>

2016). We chose to base our relatedness model on a web-based corpus instead of a news-based one to favor discovering related words from various domains, which would be perceived as more humorous.

If the target  $t$  is an entity, we search the first resources (i.e. the NOC list and the expanded properties) to collect its top  $k$  stereotypical properties. In case no available knowledge regarding the entity existed, we attempt to acquire the top  $k$  related words to the rest of the potential targets (subjects and nouns, respectively) using the second resource (i.e. the semantic relatedness model). In our case, we empirically set  $k$  to 100 to allow diversity and reduce noisy relations, while ensuring the descriptiveness of the words to the target.

To be able to place the target in a humorous light, we only regard the descriptive words that describe it negatively, which is determined by employing a polarity classifier provided by Akbik, Blythe, and Vollgraf (2018). Lastly, the connection of the replacement word to the target is assessed based on the semantic relatedness between the replacement word and the target's negative descriptions. We desire to maximize such connections to encourage replacements that are associated with the target from a negative angle.

### Generation and picking out the best candidate

We use the Humicroedit dataset of headlines published by Hossain, Krumm, and Gamon (2019) as the source of original headlines. Furthermore, the dataset contains edits performed by humans to make the headlines humorous along with a score indicating how humorous they were when perceived by other people on a scale from 0 to 3. The motivation for using this dataset is that the editors were required to make a single change to either a verb or a noun in the head-

id=corpora



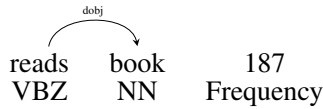


Figure 2: An example of a syntactical relation in the repository of grammatical relations (Alnajjar 2018) along with its frequency.

line to make it humorous, which focuses the scope when modeling such a process computationally.

In our generation method, we only consider headlines where the original word that is selected to be replaced is parsed as either a noun or a verb using spaCy and is a single token (i.e. ignoring cases such as “Illegal Immigrants”). The rationale behind is to reduce misparsing errors and concentrate on a single-word changes.

For an original headline  $h_o$  with its selected word to be replaced  $w_o$ , our method converts it into a humorous one  $h_h$  by replacing  $w_o$  with another word  $w_h$  as follows. It begins by acquiring replacement candidates  $C$  that fit the syntactical position of the selected word  $w_o$  by querying a massive syntactical repository of grammatical relations that have a frequency greater than 50 in a web-based corpus (Alnajjar 2018) (see Figure 2 for an example of a grammatical relation in the repository). By considering candidates that are apt to the existing syntactical relations in the headline, we ensure that the new replacement has syntactic cohesion and suits the grammatical context.

To illustrate how the method works, let’s consider the headline  $h_o$  = “City halls and landmarks turn green in support of Paris *climate* deal” as an example, where the word to replace  $w_o$  = *climate*. After parsing this headline, we find that the to-be-replaced word  $w_o$  is a noun (NN) and has a dependency (compound) on the word *deal* (NN). We query the syntactical repertory to find potential replacements that suit this relation, which yields 58 candidate replacements ( $C$  = {‘loan’, ‘business’, ‘cash’, ‘oil’, ‘holiday’, ‘peace’, ‘content’, ‘drug’ ... etc}).

In the next phase, the method removes the original word  $w_o$  from the candidates if it existed and prunes out any candidate word in  $C$  that is not identified as concrete (i.e. having a concreteness score greater or equal to 3 based on (Brysbart, Warriner, and Kuperman 2014)). As a result, candidate words such as ‘peace’ and ‘content’ in the earlier example are removed resulting in a total of 34 candidates. If there is more than 500 replacement candidates (e.g. in situations where the token to replace is a verb and is the root of the phrase), we randomly select 500 candidates in  $C$  to be examined. This is performed to reduce the search space that the method will traverse and to efficiently discover local optimal solutions as there is no particular global optimal solution for the task we are addressing.

Replacement candidates are then evaluated on the four humour aspects we are modeling, which are 1) prosody, 2) concreteness score, 3) inverted (i.e. minimized) semantic similarity between the original word  $w_o$  and the candidate

$c$ , and 4) the semantic relation between the candidate  $c$  and the negative words of the selected target  $t$ . As we are dealing with multiple criteria for modeling humour, we adopt a non-dominant multi-objective sorting approach (Deb et al. 2000) to find and select candidates in the Pareto front. Additionally, applying a non-dominant sorting for creative tasks (e.g. generating humour) increases the chances of finding balanced and diverse solutions that are more likely to be deemed good (Alnajjar, Hadaytullah, and Toivonen 2018).

Applying the evaluation and the non-dominant sorting on the example headline, the method highlights candidates such as ‘cash’, ‘meal’, ‘drug’ to be chosen as replacements. For the same example, the original word *climate* was replaced with *marijuana* by a human editor in the Humicroedit dataset. Interestingly, *marijuana* is a *drug* and our method was able to suggest it.

## Results and Evaluation

To evaluate our method, we randomly select 83 headlines from the Humicroedit dataset that meet our criteria specified earlier. For each headline, we request our method to produce humorous alternatives, ranked by the non-dominant sorting, out of which we randomly select 3 to be evaluated from the top humorous headlines.

Table 1 shows some of the headlines generated by our approach. The humorous replacement word is marked in bold. The original word and the replacement word suggested by a human from the corpus are shown in their respective columns.

We conduct our evaluation on Figure-Eight<sup>3</sup>, which is a crowd-sourcing platform that assigns paid reviewers for tasks such as questionnaires. We evaluate all the 3 variations produced by our system for the 83 headlines, showing the original headline as well. In addition, we evaluate the human edits for the same headlines from the dataset. The reviewers were not told they were evaluating computer generated humor, as the mere knowledge of a computer being an author of a creative artefact is known to provoke a bias towards seeing the generated output in a more negative light (see (Colton, Wiggins, and others 2012)).

We asked five people to rate the headlines based on the following questions:

1. The altered headline is humorous.
2. The altered word is surprising.
3. The altered word fits into the headline.
4. The altered word is concrete.
5. The joke of the headline makes fun of a person or a group of people (also known as the target of the joke).
6. The altered word shows the target in a negative light.
7. The altered word is a pun of the original word.

We evaluate the first two questions on the scale from 0 to 3 (*Not funny*, *Slightly funny*, *Moderately funny* and *Funny*). Or surprising in the case of the Q2) similarly to the questions for humor in Hossain et al. (2017). The rest of the

<sup>3</sup><https://www.figure-eight.com/>

| Humorous headline by our system  | Original word | Human replacement |
|--|---------------|-------------------|
| Thieves carry out elaborate van heist to steal millions in <b>cereal</b> , Swiss police say                    | cash          | blouses           |
| Trump <b>eats</b> the wrong Lee Greenwood on Twitter   | tags          | woos              |
| 'I was very angry' at Trump, says Myeshia Johnson, widow of fallen <b>sock</b>                                 | soldier       | cake              |
| Trump Tried To <b>Climb</b> Heather Heyer's Mother During Funeral: 'I Have Not And Now I Will Not' Talk To Him | call          | proposition       |
| U.S. says Turkey is helping ISIS by <b>Combing</b> Kurds in Syria  | bombing       | feeding           |

Table 1: Examples of generated headlines.

questions are presented as yes/no questions. The sixth question is only visible if the fifth question has been answered to affirmatively.

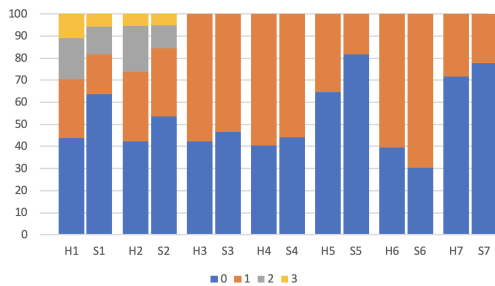


Figure 3: Percentages for each evaluation question. H marking human authored headlines, and S computer authored ones.

Figure 3 shows the percentages of the result for each question from the human evaluation. The results for the human edited titles (H) and the ones produced by our method (S) are shown side by side. From the question 3 onward, 0 marks negative and 1 affirmative answer. All in all, our system scored slightly lower on the questions than real people, which is to be expected due to the difficulty of the problem. However, our system got slightly better results in the question number 6, which means, that when the system had recognizably picked a target, it managed to convey negativity towards the target on a level comparable to a real human.

In terms of humor, our system managed to produce at least slightly humorous headlines 36% of the time, whereas people produced at least slightly humorous headlines 56% of the time. In comparison, for a recent pun generator, (He, Peng, and Liang 2019) report a success rate of 31% for their system according to a human evaluation, to put our results in a computational perspective.

Table 2 shows the results from another perspective. The *score* row shows the results for human authored titles in the original publication (Hossain et al. 2017), whereas the *human* row shows the results for the very same titles in our evaluation. The *max* shows the average of the best scoring generated headline out of the 3 ones produced for each

original headline, and *min* shows the average of the worst headline in the triplets. *Avg* is the average of the scores for all the generated headlines.

By looking at the results this way, we can see that at best, our method can produce humor comparable to real humans in the scale of funniness, with a higher amount of surprise, better aptness of the replacement word to the context, higher level of concreteness, higher negativity towards the target and higher level of puniness, falling shorter only in the case of having a perceivable target for the joke in the headline. Focusing on the best scoring individuals might sound like giving too good a picture of the performance of the system, however, they set the upper boundary for the performance of the system. This being said, with the exact same method, better results could be obtained in the future by developing a better way for ranking the humorous headline candidates output by the system.

By considering the headlines produced by our method that have the maximum score for an original headline, we see that 47 of them were credited as humorous (i.e. having a score  $\geq 1$ ) out for the 83 original title. On the other hand, 43 of the human generated were considered humorous.

In the following analysis, we aim to evaluate the different criteria considered in our method for modeling humour. In terms of prosody, we look at the number of times a headline was considered to be punny by people with respect to our method's score on the prosody dimension. Overall, 22% of the generated headlines were considered to have a pun in relation to the original word. Out of these headlines, 88% of them were evaluated positively on the prosody dimension by our system. This indicates that the method exhibited capability of assessing the sound similarity and punniness to the original word.

For the concreteness, we are considering concrete words defined in (Brysbaert, Warriner, and Kuperman 2014) as candidates. As a result, we expected to have headlines produced by the method score high on the fourth question. Contrary, only 56% of them were deemed concrete. This indicates that a more robust model is required to model the concreteness of terms.

By observing Figure 3, we notice that 46% of headlines suggested by our method are considered surprising (i.e. scoring at 1, on average). As we are using a word embeddings model, it is difficult to come up with a semantic

|       | Q1      |      | Q2      |      | Q3      |      | Q4      |      | Q5      |      | Q6      |      | Q7      |      |
|-------|---------|------|---------|------|---------|------|---------|------|---------|------|---------|------|---------|------|
|       | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ | $\mu_x$ | $SD$ |
| score | 0,99    | 0,62 | -       | -    | -       | -    | -       | -    | -       | -    | -       | -    | -       | -    |
| human | 0,97    | 0,49 | 0,89    | 0,41 | 0,58    | 0,23 | 0,6     | 0,2  | 0,36    | 0,23 | 0,61    | 0,41 | 0,28    | 0,18 |
| max   | 0,97    | 0,45 | 0,98    | 0,36 | 0,69    | 0,19 | 0,7     | 0,15 | 0,33    | 0,17 | 0,86    | 0,34 | 0,35    | 0,17 |
| avg   | 0,6     | 0,28 | 0,67    | 0,23 | 0,53    | 0,15 | 0,56    | 0,11 | 0,19    | 0,12 | 0,6     | 0,31 | 0,22    | 0,11 |
| min   | 0,28    | 0,27 | 0,37    | 0,24 | 0,36    | 0,18 | 0,4     | 0,16 | 0,06    | 0,12 | 0,27    | 0,39 | 0,1     | 0,11 |

Table 2: Mean and standard deviation of altered headlines by humans and our method.

similarity threshold that separates similar words from non-similar ones, especially for modeling surprisingness. Therefore, we test the scores assigned by the models on three thresholds of similarity (0.3, 0.2 and 0.1) with respect to the headlines viewed as surprising by online people. Out of the 46% surprising headlines, 98%, 84% and 40% headlines are considered to be dissimilar by the semantic model by using the three above mentioned thresholds. This indicates that minimizing the semantic similarity increases surprise to a degree, after which lowering the similarity results in a lower surprise.

Lastly, we perform the same analysis regarding the connection between the replacement word and the selected target with respect to question five and six. 75% of the time, our function scored positively on headlines evaluated as making fun of a target. Out of which, 77% were correctly seen as negative by the method with respect to Q6.

## Discussion

As the best headlines produced by our system for each original headline can, on the average, reach to a human level in terms of most of the factors measured by our evaluation, an immediate future direction for our research is to develop a better ranking mechanism to reach to the maximum capacity of our system. Perhaps such ranking could be learned by training an LSTM classifier on humor annotated corpora such as the one used in this paper or the one proposed by (West and Horvitz 2019).

For surprise, we opted for a rather modest approach by assimilating it to an inverse semantic similarity to the original word. However, different metrics have been proposed to model this phenomenon, such as a neural network based composer-audience model (Bunescu and Uduchi 2019) or probabilistically modelling the likelihood of a certain word occurring in a given context (see (Degaetano-Ortlieb and Piper 2019)).

The particularly low score on the concreteness highlights the inadequacy of using an annotated lexicon for its assessment. Perhaps, in the future, concreteness could be modelled in a more robust context dependent way. Previous work (Naumann, Frassinelli, and im Walde 2018) exists showing differences in the distributional representations of concrete and abstract words. As word embedding models are based on the distributional hypothesis, this discovery could be exploited for a context dependent classification by using context-aware word embeddings.

If the method was to be used as a tool for assisting journalists when composing news articles, the fact that employing computational methods for headline generation might result in offensive headlines (see (Alnajjar, Leppänen, and Toivonen 2019)) has to be taken into account. Our humor model maximizes the negative relation to its target, which might be considered as an insult, if understood in a wrong, non humorous fashion.

Our current approach focuses on English, in the future, we are interested in using our method for other languages as well such as Finnish. This would require a more robust surface realization method to deal with morphology more complex than that of English (Hämäläinen and Rueter 2018). There is already a similar semantic database available for Finnish (Hämäläinen 2018) as the one we used for English, which greatly facilitates a multilingual port of our method.

## Conclusions

We have presented a method for generating humorous headlines that in its current state, falls behind the human level humor. Nevertheless the results reach to a comparable level with an existing neural based method. The method proposed by us has the potential of reaching to a human level humor generation in the limited domain task of altering a word in an existing headline if a better ranking mechanism for its output was introduced.

The evaluation and analysis we conducted on the results has revealed several features which can be modelled better in the future to improve our method. As we have gathered human judgements for headlines generated by our system for original headlines that are based on an existing humor annotated corpus, we are releasing our evaluation results and the generated titles<sup>4</sup> in the same format as the corpus we used so that our data can be easily used in research dealing with the existing dataset.

## Acknowledgments

This work has been partially supported by the European Union's Horizon 2020 programme under grant 825153 (Embeddia).

## References

Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018*,

<sup>4</sup><https://zenodo.org/record/4976481>

27th International Conference on Computational Linguistics, 1638–1649.

Alnajjar, K., and Härmäläinen, M. 2018. A master-apprentice approach to automatic creation of culturally satirical movie titles. In *Proceedings of the 11th International Conference on Natural Language Generation*, 274–283.

Alnajjar, K.; Härmäläinen, M.; Chen, H.; and Toivonen, H. 2017. Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the 8th International Conference on Computational Creativity*, 25–32. Atlanta, United States: Georgia Institute of Technology.

Alnajjar, K.; Hadaytullah, H.; and Toivonen, H. 2018. “Talent, Skill and Support.” A method for automatic creation of slogans. In *Proceedings of the 9th International Conference on Computational Creativity (ICCC 2018)*, 88–95. Salamanca, Spain: Association for Computational Creativity.

Alnajjar, K.; Leppänen, L.; and Toivonen, H. 2019. No time like the present: Methods for generating colourful and factual multilingual news headlines. In *The 10th International Conference on Computational Creativity*, 258–265. Association for Computational Creativity.

Alnajjar, K. 2018. The 12 million most frequent English grammatical relations and their frequencies. <https://doi.org/10.5281/zenodo.1255800>.

Attardo, S., and Raskin, V. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor-International Journal of Humor Research* 4(3-4):293–348.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Brownell, H. H.; Michel, D.; Powelson, J.; and Gardner, H. 1983. Surprise but not coherence: Sensitivity to verbal humor in right-hemisphere patients. *Brain and language* 18(1):20–27.

Brysbaert, M.; Warriner, A. B.; and Kuperman, V. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods* 46(3):904–911.

Bunescu, R. C., and Uduehi, O. O. 2019. Learning to Surprise: A Composer-Audience Architecture. In *Proceedings of the Tenth International Conference on Computational Creativity*, 41–48.

Burroway, J. 2007. *Imaginative Writing: The Elements of Craft*. Pearson.

Cai, Y.; Li, Y.; and Wan, X. 2018. Sense-aware neural models for pun location in texts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 546–551. Melbourne, Australia: Association for Computational Linguistics.

Cattle, A., and Ma, X. 2018. Recognizing humour using word associations and humour anchor extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1849–1858. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Colton, S.; Wiggins, G. A.; et al. 2012. Computational creativity: The final frontier? In *Ecai*, volume 12, 21–26. Montpellier.

Costa, D.; Oliveira, H. G.; and Pinto, A. M. 2015. In reality there are as many religions as there are papers—first steps towards the generation of internet memes. In *ICCC*, 300–307.

Deb, K.; Agrawal, S.; Pratap, A.; and Meyarivan, T. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In Schoenauer, M.; Deb, K.; Rudolph, G.; Yao, X.; Lutton, E.; Merelo, J. J.; and Schwefel, H.-P., eds., *Parallel Problem Solving from Nature PPSN VI*, 849–858. Berlin, Heidelberg: Springer Berlin Heidelberg.

Degaetano-Ortlieb, S., and Piper, A. 2019. The scientization of literary study. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 18–28. Minneapolis, USA: Association for Computational Linguistics.

Gatti, L.; Özbal, G.; Guerini, M.; Stock, O.; and Strappavara, C. 2015. Slogans are not forever: adapting linguistic expressions to the news. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Härmäläinen, M., and Rueter, J. 2018. Development of an open source natural language generation tool for finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, 51–58.

Härmäläinen, M. 2018. Extracting a semantic database with syntactic relations for finnish to boost resources for endangered uralic languages. *The Proceedings of Logic and Engineering of Natural Language Semantics 15 (LENLS15)*.

He, H.; Peng, N.; and Liang, P. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1734–1744. Minneapolis, Minnesota: Association for Computational Linguistics.

Hong, B. A., and Ong, E. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 24–31. Association for Computational Linguistics.

Honnibal, M., and Montani, I. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hossain, N.; Krumm, J.; Vanderwende, L.; Horvitz, E.; and Kautz, H. 2017. Filling the blanks (hint: plural noun) for mad Libs humor. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 638–647. Copenhagen, Denmark: Association for Computational Linguistics.

Hossain, N.; Krumm, J.; and Gamon, M. 2019. “president vows to cut <taxes> hair”: Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human*



- Language Technologies, Volume 1 (Long and Short Papers)*, 133–142. Minneapolis, Minnesota: Association for Computational Linguistics.
- Hämäläinen, M., and Honkela, T. 2019. Co-operation as an asymmetric form of human-computer creativity. case: Peace machine. In *Proceedings of the 1st Workshop on NLP for ConvAI*.
- Koestler, A. 1964. *The act of creation*. London Hutchinson.
- Krikmann, A. 2006. Contemporary linguistic theories of humour. *Folklore: Electronic journal of folklore* (33):27–58.
- Lynch, G. 2015. Every word you set: Simulating the cognitive process of linguistic creativity with the pundit system. *International Journal of Mind Brain and Cognition* 6(1-1).
- Naumann, D.; Frassinelli, D.; and im Walde, S. S. 2018. Quantitative semantic variation in the contexts of concrete and abstract words. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 76–85.
- Nesterenko, L. 2016. Building a system for stock news generation in Russian. In *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)*, 37–40. Edinburgh, Scotland: Association for Computational Linguistics.
- Raskin, V. 1985. *Semantic Mechanisms of Humor*. Springer Science & Business Media.
- Ritchie, G. 2005. Computational mechanisms for pun generation. In *Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05)*.
- Ross, M. D.; Owren, M. J.; and Zimmermann, E. 2010. The evolution of laughter in great apes and humans. *Communicative & integrative biology* 3(2):191–194.
- Sane, S. R.; Tripathi, S.; Sane, K. R.; and Mamidi, R. 2019. Deep learning techniques for humor detection in Hindi-English code-mixed tweets. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 57–61. Minneapolis, USA: Association for Computational Linguistics.
- Valitutti, A.; Toivonen, H.; Doucet, A.; and Toivanen, J. M. 2013. “let everything turn well in your wife”: Generation of adult humor using lexical constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 243–248.
- Veale, T. 2004. Incongruity in humor: Root cause or epiphenomenon? *Humor: International Journal of Humor Research* 17(4):419–428.
- Veale, T. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, 34–41. San Diego, California: Association for Computational Linguistics.
- West, R., and Horvitz, E. 2019. Reverse-engineering satire, or “paper on computational humor accepted despite making serious advances”. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.
- Winters, T.; Nys, V.; and De Schreye, D. 2019. Towards a General Framework for Humor Generation from Rated Examples. In *Proceedings of the Tenth International Conference on Computational Creativity*, 274–281.
- 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 7th International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.
- Yang, D.; Lavie, A.; Dyer, C.; and Hovy, E. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2367–2376.
- Yao, J.-g.; Zhang, J.; Wan, X.; and Xiao, J. 2017. Content selection for real-time sports news construction from commentary texts. In *Proceedings of the 10th International Conference on Natural Language Generation*, 31–40. Santiago de Compostela, Spain: Association for Computational Linguistics.
- Yu, Z.; Tan, J.; and Wan, X. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1650–1660.
- Zou, Y., and Lu, W. 2019. Joint detection and location of English puns. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2117–2123. Minneapolis, Minnesota: Association for Computational Linguistics.

# Appendix D: Creative Language Generation in a Society of Engagement and Reflection

## Creative Language Generation in a Society of Engagement and Reflection

George A. Wright and Matthew Purver

Cognitive Science Research Group  
School of Electronic Engineering and Computer Science  
Queen Mary University of London  
{george.a.wright, m.purver}@qmul.ac.uk

### Abstract

Many existing models of narrative and language generation use rigid sequences of steps which are cognitively implausible and limit creativity. Iterative models based on Sharples' *cycle of engagement and reflection* improve on this by incorporating self-evaluation but still have a rigid arrangement of parts. This paper outlines how a multi-agent approach could be used to break apart the cycle into a more fluid *society of engagement and reflection*, whose constituent agents interact with one another to produce a text. Our approach is to work in a simple domain in order to focus on the underlying processes, and to avoid the ELIZA effect during evaluation.

### Introduction

Narrative is how humans make sense of the world. A model of narrative generation is thus an important strand in the development of intelligent and creative machines. But, much AI and CC work on narrative generation focuses on efficient yet rigid generation of textual summaries and/or the generation of stories and scenarios in an interesting, literary domain. There tends to be less focus on the processes that take place in a human mind during the creation of a narrative text. This paper outlines first steps towards a model based on the interactions of micro-agents which should approximate theories of cognition such as Minsky's (1986) *Society of Mind*.

### The Question of Architecture

Many models of narrative and language generation use a fixed sequence of discrete steps. This is best exemplified by the data-to-text pipelines used for summarizing structured data, although neural architectures also tend to be unidirectional and run in a fixed order. The pipeline approach has been applied to many tasks, including, recently, to the description of election results (Leppänen et al. 2017). Reiter (2007) divides the data-to-text pipeline into four stages:

**Signal Analysis** A search for patterns in the data.

**Data Interpretation** Identification of "messages" from the patterns and relations between messages.

**Document Planning** Selection of messages and arrangement into a rhetorical structure.

**Microplanning and Realization** Generation of natural language text.

The stages of the pipeline explain the processes a human goes through when describing data. Indeed the work of Reiter and his colleagues is (at least in part) inspired by observations of humans (Yu et al. 2006), but the fixed, unidirectional arrangement of the processes is not realistic.

Greater realism is offered by Sharples' (1998) *cycle of engagement and reflection*, partly implemented in MEXICA (Pérez y Pérez and Sharples 2001), which is divided into stages slightly analogous to those in Reiter's pipeline:

**Contemplate** Form ideas ( $\approx$  Signal Analysis + Data Interpretation).

**Specify** Select and organize ideas ( $\approx$  Document Planning).

**Generate** Produce text ( $\approx$  Microplanning and Realization).

**Interpret** Review and interpret generated text.

The *generate* stage belongs to *engagement*, the others to *reflection*. The cycle restarts after interpretation, allowing for a consequent re-working of the text. This is more in tune with evidence from psychology and neuroscience that language production and comprehension are intertwined (Pickering and Garrod 2013). But large, self-encapsulated modules in fixed positions cannot fully account for this intertwining, nor for the fluidity and spontaneity we expect from what Fauconnier and Turner (2002, p321) term the "bubble chamber of the brain". This is the case with many models, even those using sophisticated techniques for each module such as neural networks (Fan, Lewis, and Dauphin 2019) or genetic algorithms (McIntyre and Lapata 2010).

**The FARG Approach** More fluidity and spontaneity occurs in the models of analogy making and creativity by Hofstadter and his Fluid Analogies Research Group which consist of thousands of small agents called *codelets* that gradually build (and sometimes destroy) structures in a workspace (Hofstadter and FARG 1995).

One of their earlier models is Copycat, which solves analogy problems of the form "if *ABC* goes to *ABD*, what does *XYZ* go to?" (Mitchell 1993). Similar methods have been applied to other areas such as music understanding (Nichols 2012) and typeface design (Rehling and Hofstadter 2004).

Copycat tends to produce more sensible solutions to problems, but when faced with an unusual situation can come up with less obvious solutions (such as *WYZ* to the above



problem). Hofstadter compares this to the way people resist “nonstandard ways of looking at situations” unless a change in circumstances warrants it (Hofstadter and FARG 1995, p240). The usual answer to an analogy problem like the one above would be to replace the last letter with its successor in the alphabet, only in the case of XYZ that is not possible, so a more outlandish approach is taken involving a reversal.

The Copycat architecture has three main components:

**The Workspace** where an initial problem is perceived and structures are built by codelets to represent groupings and analogical mappings. The workspace has a *temperature* indicating the coherence of its structures.

**The Slipnet** a semantic network whose nodes spread activation and slip towards and away from one another according to the current context. Active nodes send *top-down* codelets to seek instances of their concept.

**The Coderack** where codelets are selected stochastically and according to their urgency. If the workspace has low coherence, selection is more random, and more open-minded *bottom-up* codelets can explore alternative paths.

In general, *top-down* codelets become more dominant over time as the temperature (non-monotonically) decreases and a single path to a solution is chosen. It is possible that a chosen path will result in a snag — in which case the temperature will increase, offending structures will be destroyed, and alternative pathways will be considered (Mitchell 1993).

Unlike the frameworks for language and narrative generation discussed above, FARGitecture does not involve a central authority directing the model through stages in a sequence: control is distributed between codelets and slipnet nodes. When more bottom-up codelets are running, the system is in a relative state of reflection (contemplating new structures and reviewing existing ones), while when more top-down codelets are running, the system is in a relative state of engagement (pursuing a particular path towards a solution). FARGitecture therefore enables a fuzzy alternation between engagement and reflection.

Copycat’s lack of central control, tendency to vary its behaviour due to stochasticity, and ability to pursue stranger solutions when circumstances allow make its architecture more cognitively plausible than other more rigid models.

### The Question of Domain

This paper outlines how ideas developed by Hofstadter and FARG (1995) could be applied to narrative generation. Their approach is to work in micro-domains so that evaluation must focus on the decisions a program makes while exploring its search space, not on any meaning inherent to the space. This is a different approach from most work in creative language generation which tends to cite Meehan (1976) as the earliest work in the field while overlooking the more modest (yet more impressive) work of Davey (1974). Whereas Meehan’s TALE-SPIN generates stories about animals living in a forest, Davey’s PROTEUS narrates games of tic-tac-toe. PROTEUS’ subject matter is boring but its use of features such as co-reference and conjunctions produces highly readable pieces of text. TALE-

SPIN, on the other hand, outputs stories as lists of self-contained pseudo-English sentences which are easy to understand but aesthetically displeasing. Work on creative language generation tends to deal in overtly literary domains. But, all language is creative: even a tic-tac-toe commentator has to make decisions about how to structure a text; how terse or detailed to be; and what words to use where.

At this early stage in the path towards creative machines, research should avoid complex, literary domains which give the impression of creativity where there is none, and first see how decisions can be made in a simpler domain of discourse. This will prevent evaluators from succumbing to the ELIZA effect — jumping to the conclusion that a machine has achieved human levels of intelligence when it really only relies on a few simple tricks. Veale (2017) shows that, when using the same method to build plot skeletons, giving characters the names of celebrities results in higher ratings for dimensions including *imagination* and *drama* than when using generic animal characters. Readers cannot help but find meaning in a text which the artificial author is oblivious to.

Following FARG and Davey, this paper outlines a proposed architecture for narrative generation intended for testing on mini-domains such as weather and board games.

Describing a day’s weather forecast involves recognizing entities such as storms and patches of warm or cold weather; tracking their movements and changes; and weaving together these threads into one linear piece of text. Certain aspects of narrative are lacking from this domain: for example, there is no need to account for characters or their motivations. But describing the weather does require many mechanisms fundamental to narration: formulating a narrative of the weather requires the ability to select interesting pieces of information; discard other pieces; find appropriate names for the entities that have been recognized; and to find a good structure for the text. There are many non-trivial issues to tackle — even in this simple domain.

Board game narration is a domain that could provide some of the other ingredients of narrative: there are characters with goals and plans (the players), and there is space for imagined counterfactuals. In some ways board games are simpler than the weather: entities in checkers and chess are discrete whereas weather patterns have fuzzy boundaries. Board games also have a clearer beginning and end.

Ultimately, an architecture that could handle both of these domains would be a good candidate for a general model of humans’ storytelling capacity. This paper focuses, for the most part, on the domain of weather.

### A Society of Engagement and Reflection

In this (yet unimplemented) architecture everything is done by codelets, including: data interpretation; arrangement of the text; language realization; evaluation of structures; and destruction of those that are no longer wanted. These tasks correspond to the modules in pipeline and cyclic architectures discussed above, but while most models perform these functions in a strict order, in this society model the tasks are broken down into small units of work which can be carried out whenever appropriate. A codelet runs not according to its position in a line-up, but due to competing data-driven

*bottom-up* pressures and conceptual and aesthetic *top-down* pressures.

Each codelet can be classed as either *bottom-up* or *top-down*. Bottom-up codelets are more open-minded, looking for anything of interest, whereas *top-down* codelets are more single-minded, looking for instances of a specific concept.

**Data Labeling and Grouping Codelets** Bottom-up data interpreting codelets access raw data in the workspace and determine the best concept with which to label it. For example, in the weather domain, a location with a temperature of 25°C may be labeled HOT. This leads to the HOT semantic network node receiving a boost in activation. Once fully activated, this node sends out top-down codelets to look for other locations that can be labeled as HOT. After a while, many of the same labels begin to appear in one region of the map and grouping codelets, recognizing the similarity, divide the map into regions corresponding to weather type.

These codelets perform a similar role to a convolutional kernel in a neural network, indeed they could each be implemented as a neural or other machine learning classifier. The benefit of using individual codelets which are run according to the urgency determined by activations in a semantic network, instead of having fixed layers in a neural network, is that they are not necessarily run unless the combination of context and top-down desires deems it necessary. For example, having recognized a pattern of interest in the north of a map, the NORTH node in the semantic network may spread activation to the SOUTH node to encourage a search for a pattern which summarizes the south. This architecture of interacting codelets allows for higher-level relational processing to be followed by a reversion to lower-level raw-data processing similar to how Yu et al (2006) found experts switch between more coarse and more detailed views when analyzing data to get “details-on-demand”. Feed-forward neural architectures and traditional pipeline architectures, on the other hand, rely on all of the data interpretation that could possibly be relevant having been done at an early stage.

**Language Generation Codelets** Several codelets perform the task of microplanning and realization.

**Phrase codelets** recognize a structure that can be transformed into a phrase. E.g. *rainy* → *It will be rainy*.

**Connective codelets** recognize two phrases which can be joined. E.g. *It will be rainy. It will be cold.* → *It will be rainy and it will be cold.*

**Deletion codelets** remove unnecessary parts of a phrase once it has been connected. E.g. *It will be rainy and it will be cold.* → *It will be rainy and cold.*

**Ordering codelets** order two or more phrases or sentences, such as in a general-to-specific order or along a dimension of a conceptual space. E.g. *It will be warm in the midlands. It will be hot in the south. It will be cold in the north* → *It will be hot in the south. It will be warm in the midlands. It will be cold in the north.*

Phrase codelets essentially apply templates. But, the aim is to limit the size of templates and allow for them to be combined, re-ordered and re-structured in order to limit repetitiveness. This is similar to the approach taken by Leppänen

et al (2017), but this architecture should allow for more diverse realizations. For example, there may be different ways to order phrases according to the most salient concepts in the context; and there may be different ways to connect phrases according to how ordinary their co-occurrence is: *hot but rainy* makes sense; *cold but rainy* does not (at least from a British perspective). The exact realisation that the architecture chooses will in part depend on its stochasticity and it will not be expected to re-produce the same text if run again.

Other codelets are also required, such as those that arrange rhetorical structure and those that pick which information to include in the text.

### A Hypothetical Example

Figure 1 is an example of a map of the weather at a point in time for the model to describe (more realistically, it should handle a sequence of maps in order to qualify as narrative). This map has four *channels*: weather type, wind (direction and speed in kph), temperature (in centigrade) and percentage probability of precipitation. Below is an example of a textual forecast it might generate.

*It will be cloudy in the north with a high chance of rain and furthermore snow in the very north. There will be dry weather in the rest of the country but there may be pockets of rain in the south. It will be sunny in western and central areas but temperatures will be mild while it will be cloudy but warm in the southeast.*

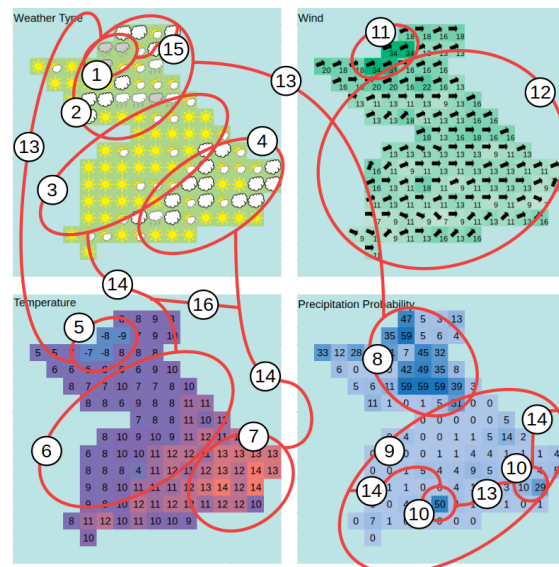


Figure 1: A four-channel map of weather in Britain with groups and relations. 1-12: Regions of similar weather; 13: AND relations; 14: BUT relations; 15: A FURTHERMORE relation; 16: A second-order AND relation. (Data from the Met Office).

At the start of the program's run bottom-up codelets search for the types of weather present on the map. Codelets also group them into regions. The ellipses in figure 1 indicate approximate regions that might be recognized.

Codelets then find relations between regions. Certain regions are recognized as being to some extent the same, for example regions 2 and 8 in the north of the country. The north's cloudy weather and high chance of rain are ordinarily co-occurring types of weather thus are connected with AND. Meanwhile the south's cloudiness and warmth are less typical so are connected by BUT. When a sub-region has a more extreme kind of weather than its parent region, for example the snow in a small part of the north, a FURTHERMORE relation is used. When a temporal sequence of events is being described, yet more relations can be recognized, such as THEN and THEREFORE. Higher-order relations are also possible: 16 shows an AND connecting two parallel BUTs.

Codelets use weather, location, and relation labels to begin forming phrases. Certain labels depend only on local concepts such as "the north", while others such as "the rest of the country" are context-sensitive.

Arrangement of the text also depends on linguistic context. For example, the sentence describing *the rest of the country* must come after the sentence describing *the north* in order for *the rest* to make sense. The sentence comparing the western and central areas and the southeast ought to come last since it is an elaboration of the sentence describing *the rest of the country*.

Codelets must recognize the importance of context and discourse relations as they arrange the final text.

### Open Questions

Many questions need to be answered in order to get this architecture working: what conceptual knowledge will the model require? Can the model be applied to board game narration and beyond? How much of the workspace context must each codelet be aware of? How will the model handle complex situations where concepts have varying relevance in different places?

This last issue, French (1995) describes as the "problem of single nodes with multiple activations". It was a major problem in his (FARGitecture based) model of analogy making between objects on a dinner table, and required a hierarchy of different contexts corresponding to different patterns of activation in the semantic network. It is likely to be an even larger problem in narrative formation, which can involve summarizing even more situations than when making a single analogy.

### Conclusion

There remain issues to be resolved in applying this style of architecture to narrative generation, but its potential for flexibility makes it an attractive line of research. Work so far has centred around the mundane domain of weather so that focus can be placed on the most fundamental issues involved in narrative and language. Future work should move into richer domains such as board game narration in order to better test the generality of the approach.

### Acknowledgments

This research has been supported by EPSRC grant EP/R513106/1 and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors' views and the Commission is not responsible for any use that may be made of the information it contains.

### References

- Davey, A. 1974. The formalisation of discourse production. PhD Thesis, University of Edinburgh.
- Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for structuring story generation. In *Proc of the 57th Annual Meeting of the ACL*, 2650–2660.
- Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.
- French, R. M. 1995. *The Subtlety of Sameness: A Theory and Computer Model of Analogy-Making*. The MIT Press.
- Hofstadter, D., and FARG. 1995. *Fluid Concepts and Creative Analogies*. Basic Books.
- Leppänen, L.; Munezero, M.; Granroth-Wilding, M.; and Toivonen, H. 2017. Data-driven news generation for automated journalism. In *Proc of the 10th INLG*, 188–197.
- McIntyre, N., and Lapata, M. 2010. Plot induction and evolutionary search for story generation. In *Proc of the 48th Annual Meeting of the ACL*, 1562–1572.
- Meehan, J. R. 1976. The metanovel: Writing stories by computer. PhD Thesis, Yale University.
- Minsky, M. 1986. *The Society of Mind*. Picador.
- Mitchell, M. 1993. *Analogy-Making as Perception: A Computer Model*. The MIT Press.
- Nichols, E. P. 2012. Musicat: A computer model of musical listening and analogy-making. PhD Thesis, Indiana University.
- Pickering, M. J., and Garrod, S. 2013. An integrated theory of language production and comprehension. *Behavioural and Brain Sciences* 36:329–392.
- Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *JETAI* 13.
- Rehling, J., and Hofstadter, D. 2004. Letter spirit: A model of visual creativity. In *Proc of the 6th ICCM*, 249–254.
- Reiter, E. 2007. An architecture for data-to-text systems. In *Proc of the 11th ENLG*, 97–104.
- Sharples, M. 1998. *How We Write: Writing as Creative Design*. Routledge.
- Veale, T. 2017. Déjà vu all over again: On the creative value of familiar elements in the telling of original tales. In *Proc of the 8th ICCM*, 245–252.
- Yu, J.; Reiter, E.; Hunter, J.; and Mellish, C. 2006. Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering* 13:25–49.

# Appendix E: Parsing Text in a Workspace for Language Generation



EasyChair Preprint

№ 6171

## Parsing Text in a Workspace for Language Generation

---

George Wright and Matthew Purver

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 27, 2021



## **Parsing Text in a Workspace for Language Generation**

George Wright<sup>1</sup> and Matthew Purver<sup>1</sup>

<sup>1</sup>Cognitive Science Research Group, School of Electronic Engineering and Computer Science,  
Queen Mary University of London

### **Author Note**

The authors declare that there no conflicts of interest with respect to this preprint.

Correspondence should be addressed to First author name and address. Email:

[george.a.wright@qmul.ac.uk](mailto:george.a.wright@qmul.ac.uk)

### **Abstract**

Processing of language by humans involves the intertwining of processes of production and comprehension. This paper describes how a cognitively inspired architecture for analogy-making can be adapted for the modeling of language generation and specifically details how a generated sentence can be parsed within a workspace in order to contribute to the program's self-monitoring and self-evaluation.

*Keywords:* workspace, codelet, natural language generation, parsing



### Parsing Text in a Workspace for Language Generation

This research aims to simulate human creativity in generating language. It adopts a cognitively inspired workspace-based architecture in which production and comprehension can interact so that self-monitoring and self-evaluation can co-occur with and influence text generation. This paper describes how sentence parsing can take place in such an architecture and how this can help text generation.

The computer program works in the domain of weather description. This provides a test-domain which is conceptually simple – only limited knowledge is required – but still linguistically challenging – information contained in many dimensions (a 2-dimensional map, multiple aspects of weather, time) must be selected and arranged into a linear text. This early iteration of the computer program works only with temperatures on a 2-dimensional map.

The architecture of the program is based on work by the Fluid Analogies Research Group such as Copycat. Copycat (Mitchell, 1993) is a model of analogy making which completes analogies between strings of the form ABC:ABD::IJK:?. Spreading activation in its concept network influences the selection of micro-agents called *codelets* which build structures in a workspace in order to solve the problem. For example, a codelet which recognizes that B is the SUCCESSOR of A will cause the SUCCESSOR concept to become more active and therefore encourage top-down codelets to seek out more examples of the SUCCESSOR relation and eventually complete the analogy accordingly. The program's lack of centralized control and stochasticity allow it to simultaneously consider multiple pathways to different solutions. Less promising pathways are gradually abandoned as a result of competition between structures in a search strategy called a *parallel terraced scan*. The program uses *computational temperature* – a

measure of the quality and coherence of the workspace – to determine how random codelet selection should be. As pathways are narrowed down, processing becomes more deterministic.

Numbo (Defays, 1995) is a related program which plays a number game in which a target number is made out of smaller numbers using addition, subtraction, and multiplication. For example, when given the target 114 and the numbers 11, 20, 7, 1, and 6, possible solutions include:

$$20 \times 6 - 7 + 1$$

$$(20 - 1) \times 6$$

Like Copycat, Numbo's permanent knowledge contains concepts which influence codelet activity in the workspace as they become activated. But, it also contains structured information in the form of *bipeds* which encode declarative knowledge of operations on landmark integers such as  $6 = 2 \times 3$  and  $100 = 5 \times 20$ . Analogy-making between prototypical operations represented in the concept network's bipeds and numbers in the workspace guides the search for a solution.

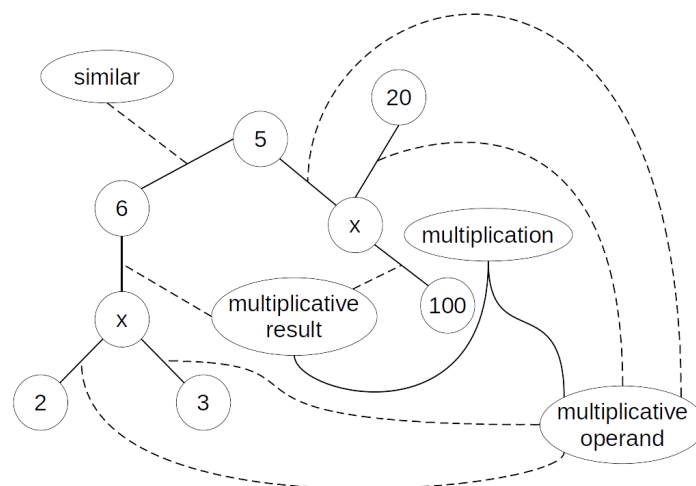


Figure 1: Part of Numbo's conceptual network with two bipeds for multiplication (Defays, 1995, p.136).

## Parsing Text in a Workspace

5

Because these architectures center on a workspace where many processes take place concurrently, they are also ideal for modeling the interleaving of production and comprehension processes which occur when humans process language (Pickering & Garrod, 2013). They should also allow for interference between production and comprehension since residual activation of a concept as a result of comprehension would make it more likely to influence production.

Gan *et al.* (1996) show how codelets operating at the level of the sentence, phrase, and word can solve the problem of ambiguous word boundaries in Chinese. But there has been little further work using this style of architecture in language processing.

### Method

The architecture centres around four components:

1. A collection of workspaces where the input, intermediate structures, and output text are worked on.
2. A collection of conceptual spaces where domain-specific concepts such as HOT and grammatical concepts such as NOUN are stored. Not all concepts are connected as part of a network as in Copycat and Numbo: temperature and location concepts are stored respectively in a TEMPERATURE and LOCATION space where a distance metric rather than explicitly instantiated connections determine similarity between concepts. The conceptual spaces therefore sit between the vector space representations described by Gärdenfors (2014) and more traditional symbolic networks.
3. The coderack, where codelets wait to be selected stochastically to enter the workspace. Each codelet is responsible for evaluating or altering workspace structures.
4. A measure of the model's satisfaction with its work so far, equivalent to 1 - *temperature* in the aforementioned programs, here called *satisfaction* to avoid confusion with weather. Lower satisfaction results in more random codelet selection so that more diverse alternatives can be explored.

Structures built by the program include *chunks* used to recognize homogeneous regions on a map; *labels* which classify items, for example a chunk could be labeled HOT and the word “hot” labeled ADJECTIVE; *relations* between two items, for example one chunk may be MORE hot than another; *correspondences* which indicate that two items, for example part of the input and an element in a frame or template are the SAME; correspondences between elements of the input

## Parsing Text in a Workspace

7

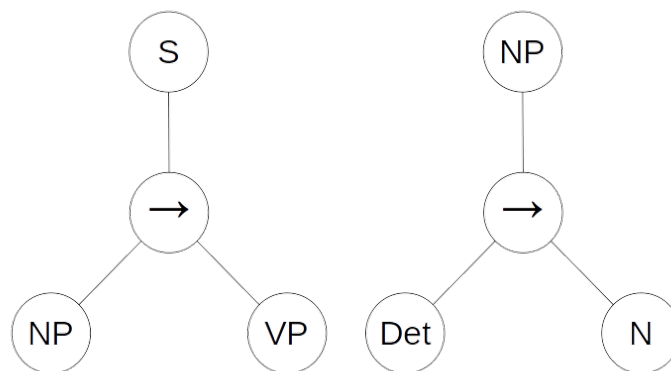
and template slots allow for the generation of *words*; parsing of words results in the creation of *phrases* which are chunks of words labeled with a grammatical role such as NOUN-PHRASE.

The program starts with a workspace containing a 2-dimensional map of the weather. As the program runs, label, chunk, and relation building codelets divide the map into areas of similar weather, classify the temperatures as COLD, WARM, *etc* and make comparisons between the temperatures. Templates containing common weather description phrases help sentence construction: correspondences connect relevant label values to relevant template slots and words are placed into an output sentence. Phrase building codelets check that the sentence is complete and an interpretation of chunks and labels is reverse-engineered out of the text. If correspondences can be built connecting the interpretation to the original input, the text is deemed adequate and is good enough to be output. Throughout the program's run multiple structures which are not necessarily consistent with one another can be built. If correspondences cannot be built between an output and the input, alternative structures and textual outputs must be selected for. This architecture makes language understanding an important and integrated part of the generative process whereas natural language generating programs have traditionally been unidirectional and modular (Gatt & Krahmer, 2018, p.82-101).

## The Program as Parser

From the program's codelets emerge macro-processes for data interpretation, language generation, and language understanding.

The (constituency) parsing components of this program are analogous to Numbo: bipeds containing mathematical operations are replaced with bipeds containing grammatical rules. Codelets gradually build a parse tree by labeling words with part-of-speech tags and chunking them together into larger phrases if they match with a rule.



*Figure 2: Numbo-style bipeds for a context-free grammar.*

This bears some resemblance to chart parsing: the workspace is essentially a chart of intermediate structures. Label and phrase evaluation and selection codelets decide which of the structures receive further attention and which are abandoned.

Evaluation codelets determine the quality of structures. Label quality is determined by the likelihood that the word is an instance of the label. Phrase quality is determined by:

- The quality of the constituent branches,
- The activation of the rule,
- The number of phrases it is contained within (how useful the phrase turned out to be in further parsing),



## Parsing Text in a Workspace

9

- The length of the phrase (this prevents low quality scores for phrases high up the parse tree).

Selection codelets choose between two competing structures, in this case alternative structures which cannot both belong in the same parse tree. Selection codelets select two competing structures and probabilistically boost the activation of the higher quality structure and dampen the activation of the lower quality structure. Lower quality structures still have some chance of being selected in case they can be used to create a better overall parse.

### Results

When treating the program as nothing but a parser using a context free grammar, out of 1000 runs, it took on average 613 codelets to parse the sentence “it is warm in the south”. The program's non-determinism allows for it to use left-recursive rules such as  $s \rightarrow s, pp$ .

In one example run, label building codelets first apply labels to words and then gradually phrase building codelets try to construct phrases. They are unable to do so until both words in a potential phrase have part-of-speech labels compatible with a rule. For example, “the” and “south” are at first labeled DET and ADJ respectively but a noun-phrase can only be created once “south” has been labeled with NOUN. As with the model in Gan *et al* (1996, p.547), processing tends to move from lower level units (words) to higher level units (phrases), but the ordering is not strict and can be interleaved.

Currently the program uses a bottom-up strategy of randomly classifying and pairing up structures to try and make phrases. Better use of the spreading activation network could improve the efficiency of the search. For example, the DET concept could spread activation to the NOUN-PHRASE concept to push the search in a more fruitful direction resulting in something more like a left-corner parser.

### Discussion

Parsing contributes to the model's language comprehension abilities, allowing it to know when a sentence is correct and complete. This should give it the ability to produce language more fluid than language based on templates alone.

For example, it could cut short a sentence such as “it is warmer in the south than the north” to “it is warmer in the south” or “it is warmer” when context allows. A better understanding of the grammar of sentences should also allow for better text manipulation when combining multiple sentences, for example deletion of repeated subjects.

Of course, for the grammatical knowledge to be made use of, it needs to exist alongside other levels of processing including at the level of semantics and discourse. Future iterations of this program must include codelets operating at these levels.

Whether or not grammatical knowledge improves the output of the program can be tested by comparing the program's behaviour with and without parsing enabled. Multiple runs of each version of the program will show the distribution of outputs it can produce as well as the length of time (or number of codelets) required to produce an answer. The quality of the outputs can also be compared by human judges.

### Comparison with Related Work

Similar work which makes use of parsing or comprehension for language production include cognitive models such as that proposed by Pickering and Garrod (2013) and work which makes use of the dynamic syntax paradigm such as Purver and Otsuka (2003).

## Parsing Text in a Workspace

12

The architecture described above bears some resemblances to that described by Pickering and Garrod with templates standing in for their forward models (impoverished, easy to compute representations) and the correspondences built between input and templates matching the comparison between the output of the production implementer and the forward model. The use of parsing for self-comprehension, however, is more similar to the internal loop of more traditional models such as Wheeldon and Levelt (1995), which Pickering and Garrod do not discount as also playing a role in self-monitoring albeit at a different level (Pickering & Garrod, 2013, p.340).

The architecture in its current form does not match well with models based on dynamic syntax since in these models, generation and parsing are one and the same process as opposed to two interleaved processes. That said, it may be worth considering dynamic syntax or other formalisms as an alternative to context-free grammar. Dynamic syntax does not prescribe a particular algorithm or architecture and since it is also a representation based on nodes and links, this architecture of codelets incrementally building structures in a workspace can be readily adapted to it.

**Acknowledgments**

Purver is partially supported by the EPSRC under grant EP/S033564/1, and by the European Union's Horizon 2020 program under grant agreement 825153 (EMBEDDIA, Cross-Lingual Embeddings for Less- Represented Languages in European News Media). The results of this publication reflect only the authors' views and the Commission is not responsible for any use that may be made of the information it contains.

### References

- Defays, D. (1995). Numbo: A Study in Cognition and Recognition. In Hofstadter D. J. (Ed.) *Fluid Concepts and Creative Analogies*. (pp. 131-154). Basic Books.
- Gan, K. W., Lua, K. T., & Palmer, M. (1996). A Statistically Emergent Approach for Language Processing: Application to Modeling Context Effects in Ambiguous Chinese Word Boundary Perception. *Computational Linguistics*, 22(4), 531-553.
- Gärdenfors, P. (2014). *The Geometry of Meaning: Semantics Based on Conceptual Spaces*. The MIT Press.
- Gatt, A., & Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications, and Evaluation. *Journal of Artificial Intelligence Research*, 61, 65-170.
- Mitchell, M. (1993). *Analogy-Making as Perception: A Computer Model*. The MIT Press.
- Pickering, M. J., & Garrod, S. (2013). An Integrated Theory of Language Production and Comprehension. *Behavioral and Brain Sciences*, 36, 329-392.
- Purver, M. and Otsuka, M. (2003). Incremental Generation by Incremental Parsing: Tactical Generation in Dynamic Syntax. In *Proceedings of the 9<sup>th</sup> European Workshop on Natural Language Generation*. (pp. 79-86).
- Wheeldon, L. R., & Levelt, W. J. M. (1995). Monitoring the Time Course of Phonological Encoding. *Journal of Memory and Language*, 34(3), 311-334.

# Appendix F: Evaluating Natural Language Descriptions Generated in a Workspace-Based Architecture

## Evaluating Natural Language Descriptions Generated in a Workspace-Based Architecture

George A. Wright<sup>1</sup>

<sup>1</sup>Cognitive Science Research Group  
School of Electronic Engineering and Computer Science  
Queen Mary University of London  
george.a.wright@qmul.ac.uk

Matthew Purver<sup>1,2</sup>

<sup>2</sup>Department of Knowledge Technologies  
Jožef Stefan Institute  
Ljubljana, Slovenia  
m.purver@qmul.ac.uk

### Abstract

This paper concerns the evaluation of a workspace architecture for generating natural language descriptions, including methods for evaluating both its output and its own self-evaluation. Herein are details of preliminary results from evaluation of an early iteration of the architecture operating in the domain of weather. The domain is not typically seen as creative, but provides a simple testbed for the architecture and evaluation methodology. The program does not yet match humans in terms of fluency of language, factual correctness, and how completely the input is described, but human judges did find the program's output easier to read than human generated texts. Planned improvements to the program also described in the paper will incorporate self-monitoring and better self-evaluation with the aim of producing descriptions that are more fluently written and more accurate.

### Introduction

This paper describes work towards a self-evaluating architecture for language generation first described in (Wright and Purver 2020) and a method for evaluating the architecture by comparing human judgements of its output with its own self-evaluation. This iteration of the architecture operates in a toy domain: making simple descriptions of temperatures on a static, two-dimensional map but serves as an initial framework on which future versions performing more ambitious tasks can be built.

### Theoretical Background

According to Fauconnier (1994), linguistic meaning is organized in mental spaces and according to Fauconnier and Turner (2002), creativity involves the projection of structures across mental spaces, often with the help of frames. Such processes cannot involve a deterministic search for an optimum, but instead a constant competition between structures evolving in a bubble chamber of mental spaces, only some of which become available to consciousness (Fauconnier and Turner 2002, p.321).

The architecture described below implements the projection of structures across spaces while making use of an enzymes-in-cytoplasm metaphor of cognition similar to that

proposed by Barrett (2005) which allows for a chaotic interaction of processes in a shared workspace or *bubble chamber*. These include processes of language production and comprehension which also interact when humans use language (Pickering and Garrod 2013). In this architecture, self-comprehension and self-evaluation are important because they help to determine which of the competing intermediate structures are used in future processing. An overall *satisfaction* score also affects how randomly processes occur. Evaluation of this architecture therefore takes into account not only the finished outputs of the program, but also its method for self-evaluation.

### The Planned Architecture

The architecture has a *bubble chamber* and a *codrack*. The bubble chamber contains a network of concepts, frames, and their instantiations spread across a number of conceptual and working spaces. These are the long- and short-term memory of the program. The best, most useful structures *bubble* to the top of the program's attention as their activation increases.

The codrack, borrowed from Copycat (1993) and related work (Hofstadter and FARG 1995) contains a collection of codelets, (small tasks to be carried out), each of which has an urgency influencing the likelihood it runs. Codelets correspond to the enzymes of Barrett's metaphor. They are selected from the codrack with a degree of randomness determined by the program's *satisfaction*, a score of the quality of active structures in the bubble chamber (a structure's quality is determined by evaluation codelets). High satisfaction leads to less random codelet selection thus more deterministic processing whereas low satisfaction leads to more randomness and opens a broader set of pathways to be explored. Self-evaluation is central to the architecture and is therefore important to consider when judging its performance.

Most codelets make a small change to the bubble chamber, for example by building a new node or link, or by changing a structure's activation. All structures, including representations of the input, parse trees, and output text are built incrementally in this manner. Codelets also change the codrack by adding a follow-up codelet. Some codelets operate exclusively on the codrack by adding or removing codelets in order to ensure that the codrack does not become empty or overcrowded.

This style of architecture shares similarities with models



based on Baars' (1997) Global Workspace Theory such as (Misztal and Indurkha 2014) which has *experts* performing tasks in a shared workspace. But, where as codelets in this architecture are restricted to performing small operations, some experts in Misztal and Indurkha's architecture such as the *metaphor expert* operate at a much higher level and perform tasks comparable in complexity to work performed by a large collective of codelets.

### Engagement and Reflection Cycles

According to Sharples (1998), the creative writing process involves a cyclic alternation between engagement (producing new ideas) and reflection (evaluating work so far). This has been implemented in models of language generation (Pérez y Pérez and Sharples 2001) as well as other models of creativity (Pérez y Pérez, de Cossio, and Guerrero 2013). The E-R model is a relatively high-level view of cognition which does not recognize the more intertwined nature of production and comprehension described by Pickering and Garrod (2013).

This architecture contains something like an engagement-reflection cycle but at multiple levels of abstraction and, due to the stochasticity of the coderack, with less rigidity.

**Codelet Cycles** Most codelets operating in the bubble chamber belong to one of four types: *suggesters*, *builders*, *evaluators*, and *selectors*.

Suggesters find an element in the input such as a temperature on a map and suggest a possible structure that can be built for that element. For example, a temperature could be labeled as HOT or in the SOUTH, two temperatures could be combined into a single chunk if they are similar, two temperatures could be connected with a MORE or LESS relation, or a SAMENESS correspondence could be recognized between a chunk in the input and an item in a frame.

Having performed a classification, a suggester codelet places a builder codelet on the coderack with an urgency matching its confidence in its suggestion. If the builder codelet is run, the relevant structure is built and the builder codelet then places an evaluator codelet on the coderack.

Evaluator codelets determine the quality of the structure according to the same classifier as the suggester. Since certain classifications can be context dependent, for example a part-of-speech label may depend on how a word is used in a sentence, the classification of a structure by the time the evaluator is run may differ from when the structure was first suggested. The evaluator assigns a quality score to the structure and then places a selector codelet on the coderack.

Selector codelets compare two competing structures, for example two incompatible labels, and boost the activation of one while depressing the activation of the other such that only one structure is likely to be used in further processing. Higher quality structures are more likely to receive a boost in activation. Selector codelets also place another suggester on the coderack thus completing a cycle at the fine-grained level of workspace structures.

If a codelet fizzles because the bubble chamber does not contain the right conditions or if a follow-up has low urgency and never runs, the cycle breaks. Meanwhile new cycles are

created as *factory* codelets add new suggesters and evaluators to the coderack so that processing does not stop prematurely.

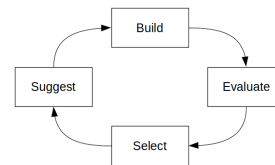


Figure 1: The lowest-level “cycle of engagement and reflection” at the level of individual nodes and links in the bubble chamber.

**View Cycles** The architecture implements the *simplex networks* of Fauconnier and Turner (2002, p.120-2), which connect elements in an input space to elements in a frame and then elements in both the input and the frame to new elements in an output space. Since this is a language generating program, the frames are templates with slots to be filled in according to the input. The output is a text which describes the original input using the template structure. Each network exists within a *view* based upon the *Worldview* of the Table-top model of analogy-making (French 1995). All structures within a view must be consistent with one another.

The architecture also uses views for self-monitoring. *Monitoring views* contain an output text, a semantic parse of the text and a set of correspondences between elements of the parse and the original input. The purpose of a monitoring view is to check that a text both makes sense and is an accurate description of at least part of the original input.

Texts which have been matched to part of the original input are made available for further processing inside higher level simplex networks using discourse frames. This allows for a recursion of simplex networks as described by Fauconnier and Turner (2002, p.151) and produces a cycle of engagement and reflection at the higher level of fragments of text which emerges from the cycles of engagement and reflection at the lower level of individual nodes and links.

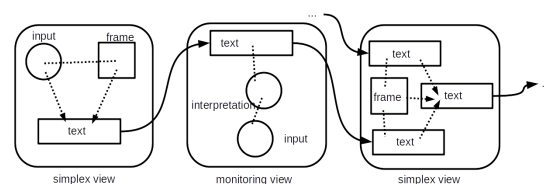


Figure 2: A higher level “cycle of engagement and reflection” at the level of pieces of text.

### The Current Implementation

The implementation of the architecture described above operates within a simple domain, describing temperatures on a map. This requires a small knowledge base and allows for focus to be placed on the mechanisms of the architecture.

Implementation is still in an early stage and lacks much of the self-monitoring provided by monitoring views.

In order to get the program to output text, a *publisher* codelet occasionally runs, which finds templates that have had their slots filled and outputs the resulting text. Current outputs are therefore short and lack discourse structure, but the evaluation of these outputs provides a base-line upon which future iterations of the model can improve.

The current implementation's satisfaction is calculated as the mean of the product of each bubble chamber structure's quality and activation. This means the satisfaction is higher when the most active structures have a high quality and lower when active structures have a low quality or high quality structures have a low activation. But, as discussed below, this results in a satisfaction score which fails to take into account a more global perspective on the bubble chamber.

### Evaluating The Program

The relatively transparent nature of the program allows it to be evaluated in a number of ways: the intermediate representations it builds when processing the input, its textual output, its understanding of its own textual output (through syntactic and semantic parses), and its satisfaction score for its output can all be seen and evaluated by external observers.

Below is described a subjective and intrinsic evaluation of outputs of the system implemented thus far - a survey which evaluated the system in isolation from any practical application and according to human value judgements. Such surveys commonly focus on two main criteria: the quality of a text, and its accuracy relative to the input (Gatt and Krahmer 2018, p.124).

### The Survey

Human subjects in the survey were asked to compare two of the program's outputs for each input. They had to answer four questions for each pair:

1. Which text is easier to understand?
2. Which text is more fluent?
3. Which text is more factually correct?
4. Which text represents the map more completely?

Respondents could answer each question in one of three ways: the first text is better than the second, the second text is better than the first, or the two texts are approximately equal.

The aim of the first two questions was to capture the linguistic quality of the texts, while the aim of the final two questions was to capture their accuracy as descriptions of the input map. Survey respondents only saw the map after the first two questions so that any inaccuracies in the description would not influence the quality score.

Human subjects had to compare two outputs rather than score them on a scale as it is unclear what the criteria are for high or low scores, especially when viewing the first few outputs from a program. Furthermore, Belz and Kow (2010) compared preference-based evaluation to score-based evaluation and found that preference-based evaluation results in less variance between respondents.

Since the computer program provides its own satisfaction score for its work, human evaluation can also be used to check if its internal measure of satisfaction matches with human judgements or if its method for calculating satisfaction could be improved. Since the program only has a single number to describe its "satisfaction", there is no one-to-one correspondence with the questions used to judge linguistic quality and factual accuracy. The score is also an absolute number rather than a preference judgement. Nevertheless, rankings based on human judgements and rankings based on the program's internal score ought roughly to align.

Methods for evaluating the creativity of computer programs commonly try to rate the novelty of outputs as well as their quality, see for example (Ritchie 2007). This is not attempted here since the domain is so simple and the outputs are so short that no output is likely to be in any way novel. It is hopefully clear though, that this architecture could in theory be applied to a more complex domain that would allow for more exciting outputs where novelty would be worth considering.

### Generation of Texts for the Survey

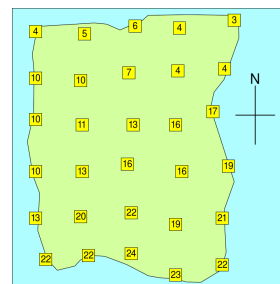


Figure 3: The first input as displayed to survey respondents. Numbers show temperatures in centigrade.

The survey was carried out using four different inputs to the program. For each input, the program was run ten times, and three outputs were randomly selected. Outputs all took the form of simple statements of fact. Added to these outputs were two human-generated descriptions which were gathered from a separate survey. For each input, one description was selected which was written with detailed, full sentences while the second description was brief and often written in note-form. At no point were the respondents told that they were evaluating machine-generated or human-generated text. The texts used for the first input were:

- A (Human) "The temperature is cold in the north but progressively warm moving south, reaching 24 degrees."
- B (Computer) "It is hot in the southeast."
- C (Computer) "It is mild in the northeast."
- D (Computer) "The north is mild."
- E (Human) "Cool in the north, warm in the south."

The purpose of including human-generated outputs was to check that respondents (on Amazon Mechanical Turk)

|   | Easiness | Fluency | Correctness | Completeness |
|---|----------|---------|-------------|--------------|
| 1 | B        | A*      | E*          | E*           |
| 2 | D        | D       | A*          | A*           |
| 3 | A*       | E*      | B           | B            |
| 4 | C        | C       | D           | D            |
| 5 | E*       | B       | C           | C            |

Table 1: Average rankings according to the pairwise preferences of survey respondents for texts describing the first input. \*Human-generated texts.

understood the task and were not pressing random buttons. A respondent who understands and pays attention to the task ought at this point broadly to prefer the human-generated texts. In future, improved iterations of the program ought to surpass the briefer note-like human-generated texts. Outputs of future iterations can also be compared to outputs of the current iteration to check if changes to the architecture result in improved results.

### Results of the Survey

The results of the survey are unsurprising in that they show that the program is overall below human-level performance, but they also highlight certain issues that should be taken into account in future evaluation.

It should first be noted that respondents of the survey did not show a high degree of agreement. The Fleiss' Kappa scores were 0.342 for ease of understanding, 0.238 for fluency, 0.484 for factual correctness, and 0.485 for completeness (to calculate Fleiss' Kappa the three possible answers to each question were treated as a category). This may in part be due to the fact that respondents had a different understanding of the questions they were being asked: future surveys should make more clear what each of these terms means, especially *correctness* and *completeness* which some respondents seemed to treat as the same. Low agreement may also have been caused by arbitrary decisions being made when similar computer outputs were compared. The survey also only had 7 respondents. In future, surveys using more respondents may result in better agreement.

Respondents on average, ranked human-generated texts above computer-generated texts along the dimensions of fluency, correctness, and completeness. But they found computer-generated texts easier to understand. A similar result was found by Reiter *et al* (2005, p.138) who found that readers preferred a computer program's weather forecasts to those written by human's due to greater consistency in the program's word choices. It is likely to be the case that more rigid and precise computer programs will always outperform humans along this dimension within small data-to-text applications, but this should be less easy to achieve in more complex domains requiring narrative or explanation. Achieving greater ease-of-understanding scores will therefore not be a priority in future work on this architecture where the aim is to achieve something closer to human-like creativity in language generation.

For the most part, no preference was shown for one text's easiness or fluency over another when two computer-

generated outputs were displayed side-by-side. This is understandable given that computer-generated outputs all followed one of two sentence patterns: the [location] is [temperature] and it is [temperature] in the [location]. Some computer-generated texts used words which did not match well with the input map and were therefore not preferred when it came to correctness and completeness.

There may have been some confounding variables which affected respondents' evaluation of the text, for example the length of the sentences being compared. Future evaluation should consider the extent to which such variables influence people's preferences.

### Evaluating the Program's Self-Evaluation

The linguistic similarity of the outputs is reflected in the computer program's satisfaction scores. The 40 runs executed for the purpose of evaluation had a mean satisfaction score of 0.704 with a standard deviation of 0.065. But, the program even had similar satisfaction scores in the 12 cases when it failed to produce an output before timing out after 30,000 codelets were run. This is because the satisfaction score is based entirely on the quality and activation of individual, low-level structures in the bubble chamber and does not take into account more global criteria for satisfaction such as the proportion of the input that has been described. It is clear that an improved metric for the satisfaction of the program is required but unfortunately it is difficult to compare different metrics when the program consistently produces similar outputs.

### Future Work

There are many improvements that can be made to the architecture, most urgent of which is the implementation of *monitoring views* in which codelets will build correspondences between the semantic parse of a text and the original input in order to check whether or not the text is factually correct and also to measure the extent to which the input has been described. This should reduce the incidence of inaccurate outputs.

The addition of discourse frames which the program can use to combine phrases and produce longer sentences should result in more fluent and complete descriptions of the input.

Furthermore, changes in higher level structures such as greater coverage of the input and improved discourse structure must be reflected in the program's satisfaction score. Future rounds of evaluation can consider alternative methods for calculating satisfaction and compare human rankings with the program's scoring of its own output.

### Conclusion

This paper has provided the outline of a planned architecture for language generation and a method for evaluating the architecture by eliciting human judgements of its output and comparing those judgements to the program's internal self-evaluation. Described in the paper is an early iteration of the architecture which lacks some of the core components required for self-monitoring and more complex discourse

structuring. The program's outputs are therefore still disappointing, but outputs of future versions of the program can be compared with its current outputs to see the extent to which greater self-monitoring improves performance.

### Acknowledgments

Purver is partially supported by the EPSRC under grant EP/S033564/1, and by the European Union's Horizon 2020 program under grant agreement 825153 (EMBEDDIA, Cross-Lingual Embeddings for Less- Represented Languages in European News Media). The results of this publication reflect only the authors' views and the Commission is not responsible for any use that may be made of the information it contains.

### References

- Baars, B. J. 1997. *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press.
- Barrett, H. C. 2005. Enzymatic computation and cognitive modularity. *Mind & Language* 20(3):259–287.
- Belz, A., and Kow, E. 2010. Comparing rating scales and preference judgements in language evaluation. In *Proceedings of the 6th International Conference on Natural Language Generation*, 7–15.
- Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.
- Fauconnier, G. 1994. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Cambridge University Press.
- French, R. M. 1995. *The Subtlety of Sameness: A Theory and Computer Model of Analogy-Making*. The MIT Press.
- Gatt, A., and Krahmer, E. 2018. Survey of the state of the art in natural language generation: Core tasks, applications, and evaluation. *Journal of Artificial Intelligence Research* 61:65–170.
- Hofstadter, D., and FARG. 1995. *Fluid Concepts and Creative Analogies*. Basic Books.
- Misztal, J., and Indurkha, B. 2014. Poetry generation system with an emotional personality. In *Proceedings of the Fifth International Conference on Computational Creativity*, 72–81.
- Mitchell, M. 1993. *Analogy-Making as Perception: A Computer Model*. The MIT Press.
- Pickering, M. J., and Garrod, S. 2013. An integrated theory of language production and comprehension. *Behavioural and Brain Sciences* 36:329–392.
- Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13.
- Pérez y Pérez, R.; de Cossío, M. G.; and Guerrero, I. 2013. A computer model for the generation of visual compositions. In *Proceedings of the Fourth International Conference on Computational Creativity*, 105–112.
- Reiter, E.; Sripada, S.; Hunter, J.; Yu, J.; and Davy, I. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence* 167(1-2):137–169.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds & Machines* 17:67–99.
- Sharples, M. 1998. *How We Write: Writing as Creative Design*. Routledge.
- Wright, G., and Purver, M. 2020. Creative language generation in a society of engagement and reflection. In *Proceedings of the 11th International Conference on Computational Creativity*, 169–172.