TheRiddlerBot
A next step on the ladder towards creative Twitter bots

Iván Guerrero[1], Ben Verhoeven[2], Francesco Barbieri[3], Pedro Martins[4],
Rafael Pérez y Pérez[5]
[1]Universidad Nacional Autónoma de México, D.F., México
[2]CLiPS Research Center, University of Antwerp, Belgium
[3]Universitat Pompeu Fabra, Barcelona, Spain
[4]CISUC, University of Coimbra, Portugal
[5]Universidad Autónoma Metropolitana, Cuajimalpa, D.F., México

Abstract

We present a computational model for the generation of a Twitter bot that aspires to be considered creative by generating riddles about celebrities and well-known characters. The riddles are created by combining information from both well-structured and poorly-structured information sources. This model has been implemented as an interactive Twitter bot (@TheRiddlerBot) that presents its outputs as contests to its followers, checks the posted answers and replies accordingly. Lastly, we present a discussion about the main attributes of a creative Twitter bot, and the remaining work for our bot to qualify as such.

Introduction

On several social networks, but especially Twitter, a new variety of users, the bots, are increasingly interacting not only with human users, but even among themselves. The first Twitter bots that appeared on the web were considered in the best case graceful, and sometimes even useful, or helpful, but they were far from being considered creative. To be creative usually relates to the generation of something novel and interesting, not only to oneself, but also to partners sharing a common background (Mayer 1999). According to this, a creative activity can be considered a social activity as well, since the environment evaluates any generational process to determine if it can be considered truly creative or not. In this sense, the environment establishes diverse constraints to any creational process, and the main challenge for an inventor resides in freeing himself from all these conventions to create something novel, interesting and yet valuable.

Novel ways of interacting inside social networks have added new and barely studied constraints to the creative process. Contests for the generation of micro-stories (Hamid 2014) - 100 words long stories -, similar to Tweet messages, or the generation of writing maps - writing prompts to inspire writers - (Maps 2015) have emerged from these new ways of interaction.

The problem that we tackle in this paper is the design and implementation of a Twitter bot that can be considered creative, focusing on missing features in the prevailing bots. The use of more realistic and diverse knowledge sources (Twitter, Facebook, Wikipedia, online news sites), evaluative mechanisms for its own outputs, and the definition of a purpose which surpasses the generation of pseudo-random messages, are examples of such omissions. The goal of our bot is to generate riddles about celebrities, formed as questions to encourage readers to assert the name of a famous character.

The rest of the paper is organized as follows. We first give an overview of the state of the art of Twitter bots, after which we give a general description of a model to automatically generate riddles and its implementation in a Twitter bot (@TheRiddlerBot). We present the results of a questionnaire where we asked people to evaluate a set of riddles. We then close with a general discussion of our proposal and our conclusions.

Related research

We now present relevant research from two different fields: riddle generation, and automatic Tweet generation. The existing theories related to the generation of riddles are not yet complete mainly because their descriptions only contemplate a subset of riddles (typically those in the question-answer format). Nevertheless, we present several approaches that provide relevant features that should be present in any riddle to be considered as such. Besides, we describe several first generation Twitter bots, Tweet-generating systems that autonomously perform useful and well-defined services (Veale 2014), that are using Twitter in diverse ways. We distinguish feeder bots, which create tons of Tweets for their followers; watcher bots, which are constantly looking for specific texts to extract information; and interactive bots, which ask followers for specific ways of communication and information sharing (Cook 2015). We describe different Twitter bots as examples of the state of the art. We will focus on both the creative aspects and unique features that are already present, as well as missing features.

Pepicello (Pepicello and Green 1984), among others, has researched riddles extensively, and described them as text fragments that employ ordinary language restricted by semiotic, aesthetic and grammatical artistic constraints. They argue that ambiguity in these descriptions is a key

aspect of a riddle, and they define three types: phonological (use of words with the same phonetic code), morphological (use of words with the same writing) and syntactic (phrases with different possible interpretations). According to this work, the goal of a riddle is to confuse the guesser by utilizing one or more of these ambiguities.

Additionally, Weiner (Weiner and De Palma 1993) defines a riddle as a language game, initiated by a question, with the goal to mislead the guesser. They describe two pragmatic mechanisms for the generation and comprehension of riddles: accessibility hierarchy and parallelism. The former relates to categorization, the capability to relate different concepts to accomplish specific goals. They describe parallelism as the tendency to remain in the same cognitive space unless a force makes us change to an alternative representation. They state that we, as humans, employ these two mechanisms to generate and comprehend riddles.

According to this work, there exist two types of concepts present in every category: context-invariant (what first comes to our minds) and context-variant (present when a relevant context appears). They state that a riddle must bring to our minds the context-invariant information to mislead the answer of the riddlee. Parallelism, in turn, helps on to generate false expectations on the part of the guesser.

JAPE - Joke Analysis and Production Engine - (Binsted 1996) is a question-answer riddle generation system. Herein, several strategies to generate riddles are described: syllable substitution, word substitution and metathesis. The first mechanism consists in confusing the syllable in a word with a similar sounding word; the second, confuses an entire word with a similar sounding word; the third, reverses the sounds of two words to suggest a similarity in meaning between two phrases.

To generate riddles, JAPE uses templates consisting of 'canned text' with slots where words or phrases are inserted. To determine which words are to be incorporated to the final riddle, the system makes use of predefined schemas, which establish relationships between words which must hold to build a joke. These schemas are manually built from previously known jokes.

In an effort to delineate novel uses of Twitter, Angelina-5 (Cook and Colton 2014) is a software for the generation of 3D games that uses a module to evaluate its textures, i.e. images utilized for decorating walls and ceilings inside the scenario, in a Twitter account (@angelinasgames). Each game has a theme, initiated by a word or phrase. Angelina-5 obtains a set of words associated to it from an English corpus, and uses them to retrieve sound effects, textures, 3D models and fonts to create a game. The bot periodically Tweets images and asks its followers to associate terms to it. These terms are collected into a repository to be further used as tags for the image.

This bot can be classified as a watcher with the goal of obtaining tags for a tweeted image from the user. The bot does not have any capabilities for analyzing the information received, given its very limited function within Angelina-5. Nevertheless, it is a functional example of how bots can receive information from humans to enhance the capabilities of a system, a desirable function to contemplate in our bot.

Flux Capacitor (Veale 2014) is a generator of well-formed and interesting character arcs (conceptual starting and ending points for a character inside a narrative). These character descriptions are defined in terms of properties, and a well-formed arc contemplates representative changes by looking for templates (such as $X becomes Y$) in Google n-grams (Brants and Franz 2006). Apart from that, relationships among properties to describe such states are retrieved from WordNet (Fellbaum 1998). The output of the bot serves the MetaphorIsMyBusiness (@MetaphorMagnet) Twitter bot to generate metaphors related to character twists in a story.

This bot has several aspects that differentiate it from first generation bots, such as its capability to deal with massive, poorly-structured knowledge databases (those lacking a well-defined format), and its purpose to create outputs surpassing the generation of pseudo-random messages. Another aspect of the bot is its high curation coefficient, the ratio of good outputs to all outputs, since the system contemplates mechanisms to evaluate its own outputs and filter those considered with low quality.

General description

We present a model for a Twitter agent with creative behaviors such as its abilities to utilize real-world, poorly-structured data sources, to evaluate its own outputs, and to interact with Twitter users. We describe as well the implementation of our model in a Twitter bot (@TheRiddlerBot) that generates creative riddles about fictional or real characters (e.g. celebrities) using cross-references from different knowledge bases.

Model description

The model consists of five main modules each subdivided in three layers (see Figure 1). Each module has a specific task ranging from the selection of a relevant celebrity, to the publication of the riddle in Twitter and tracing the answers of the followers. Besides, a layered structure of the system provides every module of tools for retrieving additional information from diverse sources, for processing the information available, and for evaluating its outputs. Now we describe the main characteristics of each module and how its tasks are distributed among the diverse layers of the model.

Character selection module This module initiates by retrieving a list of celebrity names from diverse knowledge bases. Some sources may have well-structured information, such as the Non-Official Characterization (NOC) list (Veale 2015), whereas others may lack this structure, such as
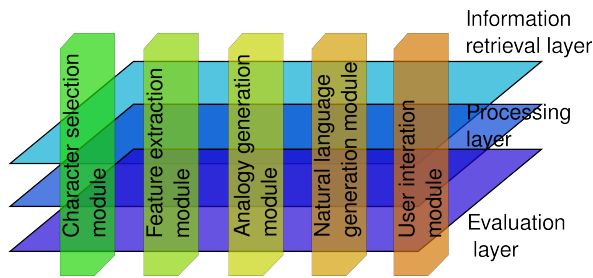
Figure 1: Model architecture

Google News, trending topics from Twitter, or public information from Facebook. This task resides inside the first layer of the module, the information retrieval layer. The data obtained is then passed to the processing layer, where one of the celebrities is selected according to diverse criteria such as his public relevance. These criteria give clues about the current importance of the celebrity due to the events he or she has recently been involved in. Finally, the evaluation layer determines if the selected character has been lately used to generate riddles, in which case it is not suitable for a new riddle. Once the character selection process finishes, the name of the celebrity is passed to the next module to look for as many facts as possible about him.

Feature Extraction Module   This second module gathers attributes about the previously selected character from both well-structured sources, such as the NOC list, and poorly-structured sources, such as Wikipedia. Furthermore, common sense knowledge bases (see the Perception dataset of the Nodebox project[1]) serve as repositories for hypernyms (super categories) of the character's attributes. These tasks are performed inside the first layer of the module. All the information obtained is then passed to the processing layer, where a subset of features is extracted according to their uniqueness and interestingness. A subset of features is considered unique if they describe only one celebrity. This evaluation is important because a riddle with unique traits is not always desirable, since it becomes easy to solve. A riddle is considered interesting when it describes a character with attributes that altogether represent relevant traits, but do not provide excessive information so that the riddle cannot be easily guessed. A set of attributes is considered relevant when the sum of the n-gram percentage of its elements, according to the Google N-gram viewer. An isolated attribute is considered to provide excessive information when its n-gram percentage is too low, and it can be considered unique. These values still need to be determined and further studies must be done to evaluate its accuracy.

Lastly, the evaluation layer determines if the subset selected has not been previously used for the same character, and that the evaluation of the attributes in previous riddles is acceptable to keep using them. These features are finally

_____
[1]http://www.nodebox.net/perception

sent to the next module to extract additional information from them.

Analogy Generation Module   The third module initiates by gathering information about similar characters according to the features of the character selected for the riddle. For this purpose, it uses information available at the NOC list as well. Then, it retrieves descriptions of analogies for the generation of relations between characters. We consider two different types of relations between a character and his attributes. Direct relations exist between a character and his features ('Diego Rivera' lived in 'Mexico', 'Tequila' is produced in 'Mexico'); higher-order relations exist between a character and a concept related to one of his features ('Diego Rivera' lived in the country where 'Tequila' is produced). For this last example, we substituted an attribute by its hypernym to create the relation ('country' is a hypernym of 'Mexico').

The information for the generation of analogies is passed to the processing layer where such analogies are created according to the attributes selected for the character. Finally, the evaluation layer determines if the mixture of attributes and analogies has been previously used to create riddles of the same character, in which case the analogies are discarded and new attributes are analyzed. With the set of features and analogies complete, the information is now passed to the next module to convert them into utterances.

Natural Language Generation Module   This module initiates with the retrieval of different types of phrasal templates for each part of the riddle (initial phrase, clues, final question). These templates are stored and retrieved from a repository specially developed for this project. A phrasal template is a previously-known sentence with slots to be further filled by specific words (Becker 1975). Each slot is commonly associated with a part-of-speech tag which allows to preserve the syntax of the sentence. Inside the processing layer, the module performs a process to select one template for each type of sentence. The selection process begins with the random selection of an initial-phrase template. Then, several clue templates are selected preventing than recently utilized templates are now repeatedly chosen. Lastly, a final-question template is picked in accordance to the first selected template.

These templates have the purpose of providing the system with a wider variety of possible generations. Once a template is selected, the slots are filled with either character's attributes, or analogy information.

User Interaction Module   The last module extracts a list of aliases for the character to be guessed. The processing layer prepares and tweets the riddle, and starts looking for responses in Twitter, which are compared against the previously obtained aliases. If there is a match, the riddle is considered to be finished. Users get points for each correct answer, which makes this system into a kind of game. When a wrong answer is detected, the user is notified and encouraged to try again. The number of incorrect answers of a

riddle can be further employed in the evaluation layer of the module. They cast light on the difficulty level of the riddle, and on the interestingness and uniqueness of the attributes employed to generate it.

System description

We have been incrementally implementing the previous model in a Twitter bot called 'TheRiddlerBot'[2]. We started out with random character selection, direct relations with tratis, and Twitter publishing capabilities. By now, we have incorporated several new features into the system, which are explained below.

The character selection process retrieves a list of celebrities from the NOC list, and randomly selects one of them. If the character has not been used to generate one of the last riddles, he is passed to the following module. The NOC list is a matrix where every row contains information about a famous character and every column is a trait, so every cell contains the value of a trait for a specific character. Several additional matrices exist where further information about specific traits, such as clothing, fictional worlds, vehicles, weapons..., can be found as part of the project. Due to its simplicity, we utilize the Pattern package in Python (De Smedt and Daelemans 2012) for a wide variety of tasks, from the feature extraction from comma-separated-values files, to the reception and sending of Tweets.

Direct relations between traits and the given character are obtained from the NOC list as well, whereas common sense knowledge is obtained from the Perception demo[3] to incorporate additional traits to the available character's knowledge.

From the list of available traits, three of them are randomly selected and evaluated to determine its interestingness and uniqueness (in the current version this evaluation is not fully implemented yet). The selected traits are considered unique when we cannot find additional celebrities inside the NOC list with the same values. To determine the interestingness of the attributes, we look for them on the character's Wikipedia page, and if they are present, we consider them relevant. If the number of characters sharing the selected values surpasses a threshold, they are not considered relevant, in such case a new subset of features is randomly selected and the generation process re-initiates. Finally, the evaluation layer looks for similar subsets for the same character in previous riddles. If this subset-character pair has been previously used, the feature extraction process re-initiates until a suitable subset is found. These features are finally sent to the next module to determine if additional relations can be obtained.

We retrieve from the additional matrices of the NOC list project information to generate higher-order relations about

fictional worlds and group affiliations. For this purpose, we look for characters who share their profession with the previously depicted character. Then, we filter out those characters who don't have any information about their fictional worlds or group affiliations. From the remaining characters, one is randomly selected to create an analogy by means of a template. We randomly depict a template from the repository available inside the system (see table 1).

Analogy type	Template
Fictional world	$<char_1>$: like $<char_2>$ in $<world_1>$
	$<char_1>$: like someone in $<world_2>$
Group affiliation	$<char_1>$: like $<char_2>$ in $<group_1>$
	$<char_1>$: like someone in $<group_2>$

Table 1: Sample analogy templates

An analogy template consist of two parts: concept, and reinterpretation of the concept, in the form $<concept>$: $<reinterpretation>$. In general, an analogy template contains redescriptions of the selected character for the riddle ($<char_1>$), in terms of a second character ($<char_2>$), and the fictional world or the group affiliation of one of the characters ($<world_i>$ or $<group_i>$). For instance, we can reinterpret the character 'The Joker', whose fictional world is 'The Dark Knight Rises', in terms of another character with the same profession, 'criminal'. In this case, we employ 'Morpheus', who can be considered a criminal in 'The Matrix', and the first template for fictional worlds, to state that 'The Joker' is like 'Morpheus' but in 'The Dark Knight Rises'. Besides, using the second template we obtain that 'The Joker' is similar to 'someone' in 'The Matrix'.

Finally, the evaluation layer determines if the mixture of attributes and analogies has been previously used to create riddles of the same character, in which case the analogies would be discarded and new attributes are selected.

The main task of the language generation module is to represent the attributes and analogies obtained from the previous step as utterances. Inside the feature extraction layer, the date of death of the character is retrieved from his Wikipedia page to determine if he is still alive or not. This information is further employed to conjugate the verbs of the generated phrases (a riddle about a deceased person is written in past tense). Some phrases require additional information of the character to present a more elaborated text, for this reason we obtain positive and negative adjectives describing the character from the NOC list. Inside the generation layer, we convert features and analogies to text. For this task we employ three different types of phrasal templates: introductory templates (see table 2), clue templates (see table 3), and final question templates (see table 4).

For the clue templates, several attributes are available for the system to select one of them. The list includes clothing, opponent, opponent activity, married partner, typical activity, vehicle and country.

Every template consists of three different types of

Type	Template
First person	I
Third person	Tell me the name of a person that

Table 2: Sample introductory templates

Feature type	Template
Group affiliation analogy	-be/VB the $<char_2>$ of $<group_1>$ -could have belonged to $<group_2>$, but do/VB not -be/VB like $<char_2>$ but be/VB not part of $<group_2>$
Fictional world analogy	-be/VB similar to someone in $<world_2>$ -be/VB the $<char_2>$ of $<world_1>$
Profession attribute	-be/VB $<value>$ -be/VB $<value>$, $<pos\_adj>$ yet $<neg\_adj>$
Opponent attribute	-do/VB not like $<value>$ -be/VB definitely not a close friend of $<value>$
Hyperonym attribute	-be/VB known as $<value>$ -be/VB $<value>$, $<pos\_adj>$ yet $<neg\_adj>$
Clothes	-have/VB been seen wearing $<value>$

Table 3: Sample clue templates

Type	Template
First person	Who might I be?
Third person	Who is this?

Table 4: Sample final question templates

Type	Value
Hypernym	'maniac', 'madman', 'criminal'
Group affiliation	'The Dark Knight Rises'
Clothes	'a purple topcoat', 'a green wig'
Pos. adjectives	'playful', 'witty', 'flamboyant', 'cunning', 'brilliant', 'creative'
Neg. adjectives	'maniacal', 'cruel', 'sadistic', 'inhuman'

Table 5: Sample attribute and analogy values for 'The Joker'

elements: words, verbs (marked with the tag VB), and slots ($<slotfiller>$). The verbs are conjugated in accordance with the type of template depicted for the introductory phrase. Afterwards, the attributes and analogies selected in the previous module are converted to text by replacing the slot fillers of the selected template with the values associated to the attributes and analogies. To conclude, the final question template is selected in accordance to the introductory-phrase template. Once the three phrases are generated in natural language, they are chunked as a riddle.

The last module tweets the generated riddle to open a new contest. To determine who wins a contest, we obtain a list of aliases for the character from his Wikipedia page. Every time a follower replies a riddle, his answer is obtained to be compared against each of the available aliases for the character, and if one of them matches, the contest is declared finished and a Tweet is published to point out the winner; if none of the aliases match, a reply to the owner is sent stating that the answer was not accurate, and the contest continues. If, after several hours, the riddle had no correct answer, a Tweet exposing the celebrity is sent, and a new contest begins.

Example of a riddle

Now, we show how to generate a riddle about 'The Joker'. Once the character selection module finishes, several attributes are obtained for the character from a variety of sources (see Table 5).

From the available attributes, a subset of three traits is selected (for this example, group, profession, and clothes), and their corresponding values are sent to the analogy module. If one of the attributes is suitable to generate analogies, the process initiates. In this case, the group attribute is used to create an analogy. We look inside the knowledge base for characters who share a hypernym (see Table 6).

Character	Group affiliation
'Fagin'	'Oliver Twist'
'John Dillinger'	'Public Enemies'
'Fredo Corleone'	'The Godfather'
'Snake Plissken'	'Escape From New York'
'Morpheus'	'The Matrix'

Table 6: Characters sharing a hypernym with 'The Joker'

With the information obtained from the previous steps, we randomly select an analogy template ($<char_1>$: like $<char_2>$ in $<group_1>$), and it is encapsulated, with the rest of the values for the natural language generation. Here, an introductory template is selected ('Tell me the name of a person that'), three clue templates are selected, one for each of the attributes or analogies employed, and a final question template is retrieved as well ('Who is this?').

The clue template for group is (be/VB the $<char_2>$ of $<group_1>$), for hypernym is (be/VB $<value>$, $<pos\_adj>$ yet $<neg\_adj>$), and for clothes is (have/VB been seen wearing $<value>$). If several values are available for an attribute, one of the is randomly picked to replace the empty slot.

Finally, we create the riddle by chunking the three templates where its slots are replaced with the corresponding values:

> Tell me the name of a person that is the Morpheus of The Dark Knight Rises, is criminal, playful yet cruel, has been seen wearing a purple topcoat. Who is this?

Model evaluation

As described above, we save all the posted riddles and their metadata (number of retweets, favorites, answers, etc.) in a database. The metadata could be used for the evaluation of the model if we assume that a riddle with more wrong answers is harder or that a riddle with a lot of favorites is better. Unfortunately, our bot is not popular enough yet, so there is very little interaction. Here are some numbers to give you an idea. At the time of writing (April 29th 2015), our bot has 57 followers. Since February 2nd 2015 (date of implementation of the database) 285 riddles were posted. Ten different users gave correct answers to 34 riddles in total.

So we decided to perform a different evaluation. We asked 86 people to each evaluate five riddles. We first asked the participants to guess the answer to the riddle. Then, we presented the correct answer and asked if they knew the person in question. The participant indicated whether he considered the quality of the riddle satisfactory and, if negative, gave us the reason why it wasn't good.

Figure 2 shows the percentage of correct answers (15.58%), and the number of known celebrities (54.19%) once the correct answer was presented.
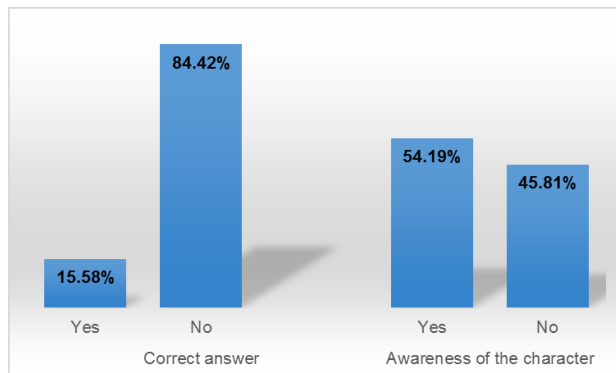

Figure 2: Results for correct answers and good descriptions

Figure 3 shows the number of riddles considered to have accurate descriptions of the characters (41.86%). When that was not the case, the main reason chosen was that the description was too vague (36.51%). Among the additional reasons given, the most recurrent was that the character was already dead and the riddle was written in present ($< 1\%$).

Finally, we present here the top 5 answered riddles, according to the number of times they appeared and the number of correct answers given to them.

Tell me the name of a person that can be found in UK, enjoys robbing from the rich, likes wearing a feathered cap. (Answer: Robin Hood).

Who is a creator, can be found in Italy, wears a paint-stained smock? (Answer: Michelangelo).
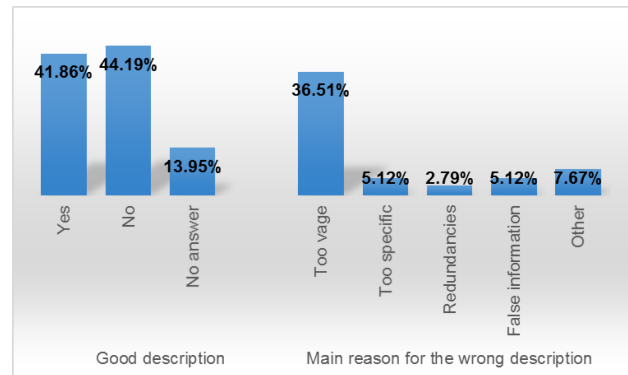

Figure 3: Results about the accuracy of the description

Who is a creative professional, pretty yet superficial, can be found in USA, enjoys monetizing celebrity status? (Answer: Paris Hilton).

Who is a religious leader, loves spreading Christianity, likes wearing sandals? (Answer: Jesus Christ).

Who is the Hermione Granger of The Simpsons, wears an orange dress, is the Timothy McGee of The Simpsons Family? (Answer: Lisa Simpson).

Discussion and future development

Relevant results were obtained from applying the questionnaire. The percentage of known celebrities once the answer was presented (54.19%) indicates that the process for the selection of celebrities should be improved. From this result we realised that almost half of the riddles could not be correctly answered because people did not have enough information about the character. One reason for this result is that owners of the NOC list (the main source for celebrities) and the riddlees were from different countries, and they did not have enough information in common.

The percentage of good descriptions of the celebrities (41.86%) represents our curation coefficient (the ratio of good outputs to all outputs), and the major cause for our descriptions to be considered wrong was its vagueness. This indicates that further work must be done to improve the interestingness of our riddles (the description of a character with relevant attributes, but without excessive information to be easily guessed). Thereby, additional mechanisms to determine the number of traits to incorporate to a riddle based on its relevance, might prevent descriptions from being too vague.

The low number of correct answers (15.58%) suggests that the complexity of the generated riddles is high. Nevertheless, by improving the character and trait selection processes will mitigate this problem.

In general terms, the current version of the system still lacks selection mechanisms relying on informed decisions. For instance, the character selection process randomly picks a celebrity from a list; the feature selection randomly chooses three character traits, despite of the final evaluation which determines whether they are good enough or not to continue with the process; most of the templates are randomly picked as well, and the values replacing the empty slots in such templates follow the same track. We consider that transforming as many random selections as possible into informed decisions will contribute in an overall increment of the final quality of the outputs generated by our bot, and will provide our model of additional traits for it to be considered creative. A key aspect to distinguish simple generation from creative generation is the curation coefficient in the outputs. To increase the number of high-quality riddles generated by our system several improvements will take place in the next release of the system.

The work presented here is a first step in building up a robust, Twitter bot that can be considered creative. For the next release we still need to improve several aspects related to intermediate output validation, and mechanisms for the automatic expansion of the current knowledge bases utilized by the system.

Despite the fact that several knowledge bases such as Conceptnet (Liu and Singh 2004) or Facebook, are not part of the project yet, the current version of the system already contains fully working mechanisms for information retrieval, and is still pending to exploit this information to generate more interesting, high quality riddles.

Conclusions

We have described a computational model to generate riddles about celebrities. It consists of modules to select a celebrity, to retrieve relevant traits to describe him, to generate analogies between his attributes and convert such descriptions into utterances, and to tweet the generated riddle and interact with Twitter users by evaluating their answers. The model presents a subdivision of each module in layers. The first layer is responsible for all the data extraction processes; the second, for processing of the information retrieved; the last, for the evaluation and validation of the generated outputs. We consider this layered approach relevant because it provides tools to enrich the intermediate outputs of every module. It contemplates the retrieval of additional information, when required, and the validation of intermediate results to achieve a higher quality in the outputs. We present an implementation of our model in a Twitter bot named 'TheRiddlerBot'. Herein, we introduce several difficulties emerging from reifying our model, such as gathering character traits, generating analogies, and generating natural language utterances.

We consider 'TheRiddlerBot' as a creative agent according to the following considerations. If we describe a creative bot in terms of its capability to deal with poorly-structured knowledge to generate something interesting and novel,

we have provided our system with such capabilities. Some authors on the field consider as essential properties for an artifact to be considered creative, novelty, quality and typicality of its outputs (Ritchie 2007). Although similar riddles can be found widespread over the literature, we consider that our system generates novel outputs since the traits employed by our implementation, considering the incorporation of analogies, make them rare to replicate. We still need to implement direct and indirect evaluations for the overall quality of the riddles, but we have sketched in this document several validation mechanisms to ensure the overall quality of our outputs. According to our definition of a riddle, questions to encourage readers to assert the name of a famous character, we argue that our outputs are typical examples of this type of queries.

According to Pérez y Pérez (2013, 2014), any output must be presented in a correct manner (coherence), generate new knowledge to the reader (interesting), and be considered new (novelty) to be creative. We verify the coherence of our riddles particularly in two stages: the analogy and natural language generation models. The analogy and phrasal templates provide the system with well-formed structures to generate complex attributes of a riddle (analogies), and to generate readable phrases written in natural language. During the evaluation layer at every module, we validate the novelty of our riddles, since at every stage of the process we ensure that the intermediate outputs have not been previously utilized. Our system considers a riddle to be interesting looking for the traits to describe a character at his Wikipedia page, and also detecting that we have not utilized the same subset on previous riddles. The first validation gives us clues about the relevance of the traits. If the reader does not know all the presented information, he will be capable of learning new qualitites of a celebrity. The second validation lets the system be certain of the uniqueness of the employed traits.

Acknowledgements

References

Becker, J. D. 1975. The phrasal lexicon. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, TINLAP '75, 60–63. Stroudsburg, PA, USA: Association for Computational Linguistics.

---

[4] http://prosecco-network.eu
[5] http://codecampcc.dei.uc.pt

Binsted, K. 1996. *Machine humour: An implemented model of puns (PhD thesis)*. University of Edinburgh.

Brants, T., and Franz, A. 2006. *Web 1T 5-gram database, Version 1*. Linguistic Data Consortium.

Cook, M., and Colton, S. 2014. Ludus ex machina: Building a 3D game designer that competes alongside humans. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Cook, M. 2015. A brief history of the future of twitterbots. Presented at the PROSECCO Code Camp on Computational Creativity.

De Smedt, T., and Daelemans, W. 2012. Pattern for python. *Journal of Machine Learning Research* 13:2031–2035.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hamid, A. 2014. Just 100 words.

Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning tool-kit. *BT Technology Journal* 22(4):211–226.

Maps, W. 2015. Writing maps.

Mayer, R. E. 1999. *Fifty years of creativity research*. Cambridge, UK: In R.J. Sternberg, Handbook of creativity.

Pepicello, W. J., and Green, T. A. 1984. *The language of riddles: new perspectives*. Columbus, USA: Ohio State University Press.

Pérez y Pérez, R., and Otoniel, O. 2013. A model for evaluating interestingness in a computer-generated plot. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Pérez y Pérez, R. 2014. The three layers evaluation model for computer-generated plots. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Veale, T. 2014. Comming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Veale, T. 2015. A game of tropes: Exploring the placebo effect in computational creativity. In *Submitted to the International Conference on Computational Creativity*.

Weiner, J. E., and De Palma, P. 1993. Some pragmatic features of lexical ambiguity and simple riddles. *Language & Communication*.