Rule-based Approaches to the Automatic Generation of Puns Based on Given Names in French

Mathieu Dehouck
Lattice, CNRS, ENS-PSL,
Université Sorbonne Nouvelle
mathieu.dehouck@cnrs.fr

Marine Delaborde
LT2D, CY Cergy Paris Université
marine.delaborde@cyu.fr

Abstract

Humor is a cornerstone of human interactions. Because puns and word plays lie in the margins of phonology, syntax and semantics, large language models struggle with their generation. In this paper, we present two versions of a tool designed to create a typical kind of French joke known as "Monsieur et Madame" jokes. We then discuss the main challenges and limitations rule-based systems face when creating this kind of puns.

1 Introduction

As of 2024, large language models are able to make jokes and explain them (Jentzsch and Kersting, 2023), but are still limited with certain types of jokes. Indeed, jokes and especially puns target a very specific aspect of language and cause a "departure from norms of usage" (Crystal, 2002) which may confuse neural models. Furthermore, some jokes are based on phonetic criteria, and most large language models are still not trained to incorporate phonetic information. This paper focuses on this kind of sound-based jokes.

Of course, jokes are plentiful on the Internet and some are even explained, for example the English Wikipedia[1] has an article about "Monsieur et Madame" jokes. Thus, large language models are likely to have encountered some jokes during their training and will likely be able to tell them back when prompted to do so. However, they have difficulties generalizing and creating new ones.

French has a templatic form of joke similar to the English "Knock-knock jokes" called the "Monsieur et Madame" jokes derived from earlier birth announcement or in memoriam jokes. In English, these jokes follow a real-life scenario (Bugheşiu, 2015) where someone knocks on a door in order to enter and when asked to identify themself, answer with a first name/family name pair that makes a punchline or the beginning of a song. In French, they also follow a fixed order but in a *one question and one answer* format. First, the punster says something along the following line: "Mr. and Mrs. X, have n children. What are their names?" where X is a made-up family name and n is a number, usually one also tells whether they are daughters or sons. Then the audience tries to solve the riddle, giving the appropriate number of first names, but often gives up eventually. The punster then reveals the actual names of the children which when said with their family name usually makes a somewhat comic sentence. The children's names must either be said independently as in (1) or in sequence as in (2). The pronunciation of the underlying sentence can be sloppy and the sloppiness is often an important part of the pun.

It is worth mentioning here that the riddle format is mostly formal and that the audience is hardly ever expected to actually find the answer. There are actually other French joke types that also rely on the riddle strategy in order to involve the audience. A side effect of this strategy, is that people who already know the answer, because they have already heard the joke before, may start to be amused before the answer is revealed.

In order to acknowledge the diversity of family structures and to make these jokes more inclusive, we actually replaced the traditional "Mr. and Mrs. X" by "Family X" in our tool. However, for the sake of simplicity, we use the traditional "Mr. and Mrs. X" in this paper.

We want to study the ability of language models to interpret, solve and then create this kind of pronunciation based jokes. However, current models are far from being able to move from written language's token space to pronunciation space and back. Even more so as French orthography is highly idiosyncratic and that French family names have broader spelling conventions than the standard

[1] https://en.wikipedia.org/wiki/Monsieur_et_Madame_jokes

Proceedings of the 1st Workshop on Computational Humor (CHum), pages 18–22

(1) Monsieur et Madame Térieur ont deux fils. Comment s'appellent-ils ?
Mr. and Mrs. Térieur have two sons. What are their names?

Alain et Alex, car, Alain Térieur (à l'intérieur) et Alex Térieur (à l'extérieur).
[a.lɛ̃ tɛʁ.jœʁ] [a l͡ɛ̃.tɛʁ.jœʁ] [a.lɛks tɛʁ.jœʁ] [a l͡e.kstɛʁ.jœʁ]
Alain and Alex, because, Alain Térieur (inside) and Alex Térieur (outside).

(2) Monsieur et Madame Cale ont trois enfants. Comment s'appellent-ils ?
Mr. and Mrs. Cale have three children. What are their names?

Anna, Lise et Mehdi, car, Anna Lise Mehdi Cale (analyse médicale).
[a.na liz me.di kal] [a.na.liz me.di.kal].
Anna, Lise and Mehdi, because, Anna Lise Mehdi Cale (medical analysis).

written language.

So as a first step, we present a series of methods used to generate first names based sentences. The different methods have their own strengths and weaknesses. The trade-off usually lies between the amount of jokes produced and the comicalness (or even meaningfulness) of jokes.

2 Why a Rule-Based approach in the AI era?

In the age of large language models, rule-based models are no longer widely used for text generation. However, in some specific tasks they may still be relevant.

2.1 State of the art

Humor can be hard to detect even for humans, so it's not surprising that it's the same for machines. However, automatic classification systems dedicated to the detection of humor, notably with the 2017 and 2021 SemEval (Potash et al., 2017; Meaney et al., 2021) focusing on tweets in English, or the HAHA challenges focusing on tweets in Spanish (Castro et al., 2018; Chiruzzo et al., 2019, 2021). Morales and Zhai (2017) use a generative language model to predict humoristic online reviews.

As for automatic generation of humor, this task is still relatively unexplored. Amin and Burghardt (2020) have produced a state-of-the-art for this task, listing 12 systems published between 1994 and 2020.

2.2 Automatic generation of puns based on names

There are currently two versions of our tool[2](Dehouck and Delaborde, 2023). The first version of the generator is based on phonetic criteria to automatically generate simple puns. The list of French terms from the English Wiktionary[3] and their pronunciation(s) (in IPA, the International Phonetic Alphabet) are used to align with a list of French names[4] and their pronunciation(s), also taken from the Wiktionary. This list of names is progressively enriched as users can enter their own name with an IPA keyboard. Alignments are made using tries (prefix trees) to find the terms whose pronunciation is compatible with the pronunciation of a name. Figure 1 shows an example of trie with pronunciations along the branches and written forms at the leaves. To find the puns, our generator identifies and shows the first names whose pronunciation in French matches the rest of the term. Then, the puns are displayed in the canonical form: a question with an answer in first(s) name(s) + last name format. This version only displays lemmas in order to avoid too much results for one pun.

The puns thus generated are displayed on a website[5]. The current version of the tool generate puns based on one first name, such as (1). Two more playful features are also being proposed. The first

[2]The first version is downloadable at the following address: https://github.com/MathieuDehouck/AliGator

[3]https://en.wiktionary.org/wiki/Category:French_terms_with_IPA_pronunciation

[4]https://en.wiktionary.org/wiki/Category:French_given_names

[5]https://tools.lattice.cnrs.fr/aligator/main.html

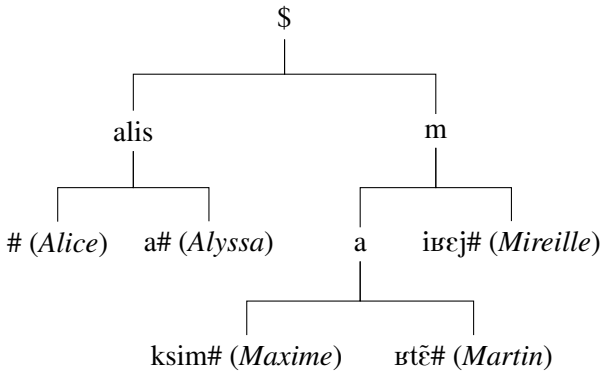

Figure 1: A trie for the given names Alice, Alyssa, Maxime, Martin and Mireille. $ represents the beginning of a string, # the end of a string and the remaining characters are plain characters. To each leaf, whose internal string contains a # character, we also associate orthographic representations for the name whose pronunciation corresponds to the path from the root to this leaf.

one, which is already available, is the generation of puns from randomly picked names. The second one is a list of the funniest puns. This one is mostly ready, but we wait to have received enough feedback from users before making it actually available.

2.3 Generation of French proper names

On our website, the user can enter a first name and ask to generate puns containing that first name[6]. If the proposed first name is not found in our database, the user can add it in IPA with its spelling variant, and thus obtain any puns that contain this name.

To display the pun in its canonical form, we created a finite-state-transducer that reads an IPA strings from the end to the beginning, applying a list of phonological rules, specific to the French language[7], that generate surnames with different levels of complexity.

For the example (1), here are some of the proper names generated: *Téryœur, Téryœurs, Théryœurt, Thérryœurd* or *Thérryœurre*. The pun generator currently uses the first form returned by the transducer, which is usually not too complex, here it would be *Théryœur*, but we are working on a process to add more diversity while avoiding forms with too many silent and double letters.

3 V2: work in progress

The first version of the tool already generates a great number of puns, but it is limited in several aspects. It is limited in term of phonological variations. And it can only generate jokes based on a fixed stock of words and expressions. In this section, we present how we plan on tackling these limitations.

3.1 Phonological relaxation

There are several reasons to allow imperfections in the phonological alignments in our tool.

As is the case for many languages, spoken and written French are quite different. At the phonological level, written French preserves a number of distinctions that are not maintained in many accents, such as the [e/ɛ] distinction in open syllables. Likewise, the close-mid/open-mid vowel distinction is strongly linked to the presence of a following coda, which leads to vowel alternation[8] in many French verbal or adjectival paradigms.

We also would like to point out that we used the transcriptions from descriptive resources as a starting point. For some of the words or first names, several pronunciations are proposed but they do not always represent all possible realizations in French. In the case of several variants, we have chosen to use the first transcription and to relax the phonological constraints in order to compensate for the absence of some variations.

To give the reader an idea of the impact of phonological relaxation, we have computed the number of puns generated by the fixed-vocabulary rule-based generator with and without the close-mid/open-mid vowel distinction. With the mid-vowel distinction enforces, out system generates 2093 puns representing 1922 lemma[9]. When the mid-vowel dis-

[6]For the moment, we only display 5 of them, for further annotation and because there are sometimes too many outputs

[7]https://fr.wiktionary.org/wiki/Annexe:Prononciation/français

[8]X-Sampa even has a dedicated / character to represent the archiphonemic status of vowels involved in this phenomenon.

[9]Note that a few lemma can appear with different first names. For example, *anaphore* can lead to *Anne Afaure* or

tinction is relaxed, the system generates 2728 puns (a 30% increase) representing 2382 lemma (a 24% increase). Some first names, such as *Eve* even go from generating 0 puns to 51 after the relaxation.

By taking this approach a step further, the release of constraints could also enable the production of more puns, such as: *Gordon Zola* [gɔʁ.dɔn zo.la], for *gorgonzola* [gɔʁ.gɔn'(d)zo.la].

As expected, relaxing phonological constraints increases the number of generated puns. However, if the pronunciations are so different that the audience has no chance of guessing the actual answer, the pun has to be really funny for the joke to work. There is a strong trade-off between the quality of the pun and the divergence from the expected pronunciation. This is in fact a place where users' feedback will indeed be valuable.

3.2 Ups and Downs of Syntax

For the second version of the generator, we want to go beyond the Wiktionary's words and expressions with IPA pronunciation and create new pun phrases using morphosyntax.

We use the Morphalou lexicon (ATILF, 2023) as a source for pronunciation and morphological information. This lexicon consists of french word forms annotated with pronunciation (in X-SAMPA) and morphological information. In this version, puns are no longer restricted to a single lemma, but span whole phrases (such as determiner + noun + adjective) and must thus respect gender and number agreement rules (using the information from Morphalou). First names are still extracted from the Wiktionary in this version.

We create morphologically annotated tries (prefix trees), associating their pronunciation with orthographic forms. In this way, we can find words whose pronunciation is compatible with the pronunciation of a given name while controlling for part-of-speech and morphological features.

Morphosyntactic backbones, that are sequences of parts-of-speech with agreement information, are extracted from Universal Dependencies treebanks (Zeman et al., 2022) in order not to generate purely random strings of words.

This method indeed generates many syntactically correct sentences which start with strings of sounds matching given names. However, they are often far from semantically acceptable, if meaningful at all, and therefore rarely make for funny jokes. For example, we generate puns such as the one in figure (3) which, while being syntactically correct, does not really make sense.

Implementing semantic consistency by using only rules is inconceivable. Lists of well-known movies, songs or expressions can also be useful, but seem insufficient.

This is the stage where large language models can actually be very useful. As they have been trained to mimic human language, they can produce semantically consistent sentences.

3.3 Annotation

Visitors are already invited to rate the puns directly on the website. They can also annotate these puns with tags such as "explicit content", "unintelligible" or a degree of humor (from *not funny* to *very funny*). For example, this is also the case on the JokerJudger platform used by Winters et al. (2018) to evaluate their system. In the future, a formal annotation campaign is envisioned for this task.

4 Conclusion

The first version of our generator uses rules to automatically produce puns based on one given name and one word or locution, respecting a well-known canonical form in French. This version is accessible online and the code is available on Github under MIT license.

We then consider three improvements for the second version. First, the phonological constraints relaxation in order to cover different variants of spoken French. Secondly, the use of a large language model for the semantic consistency of the sentence generation. Third, the annotation campaign for the degree of humor.

Limitations

Humor generation is a complex task and as we have discussed earlier in this paper our tools have a number of limitations.

The first version of our tool uses a fixed vocabulary of words and expressions, so it is limited in this direction.

The second version generates new phrases and is thus less limited, but has a semantic consistency problem.

Ethical Considerations

We do not see any direct way this work could be used unethically.

Anna Faure.

(3) Monsieur et Madame Golaux ont deux enfants. Comment s'appellent-ils ?
Mr. and Mrs. Golaux have two children. What are their names?

Laura et Valérie, car, Laura Valérie Golaux (l'or avalé rigolo).
[lo.ʁa va.le.ʁi gɔ.lo] [l ɔʁ a.va.le ʁi.gɔ.lo].
Laura and Valérie, because, Laura Valérie Golaux (the swallowed funny gold).

We actually work on making our website and tools more inclusive as we mentioned in the introduction, by removing the pun presentation from its traditional family setting.

However, we should mention that both versions of our tools are currently semantically unfiltered. For example, the first version uses any French word or expression that is annotated with its pronunciation in the Wiktionary, as source for puns. This could include all sorts of offensive terms, including and not limited to, racial, sexual or disability based slurs.

We are working on this problem. One way would be to parse the whole Wiktionary article to see if any indication is given about the offensive or derogatory nature of a word.

Since we are aware of this problem, users can actually tag puns as being offensive, if it ever was the case that our tools generated such puns.

References

Miriam Amin and Manuel Burghardt. 2020. A survey on approaches to computational humor generation. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41, Online. International Committee on Computational Linguistics.

ATILF. 2023. Morphalou. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Alina Bugheşiu. 2015. Knock-knock jokes, proper names, and referentiality. *BAS British and American Studies*, (21):183–191.

Santiago Castro, Luis Chiruzzo, and Aiala Rosá. 2018. Overview of the haha task: Humor analysis based on human annotation at ibereval 2018. In *IberEval@ SEPLN*, pages 187–194.

Luis Chiruzzo, Santiago Castro, Mathias Etcheverry, Diego Garat, Juan José Prada, and Aiala Rosá. 2019. Overview of haha at iberlef 2019: Humor analysis based on human annotation. In *IberLEF@ SEPLN*, pages 132–144.

Luis Chiruzzo, Santiago Castro, Santiago Góngora, Aiala Rosá, JA Meaney, and Rada Mihalcea. 2021. Overview of haha at iberlef 2021: Detecting, rating and analyzing humor in spanish. *Procesamiento del Lenguaje Natural*, 67:257–268.

David Crystal. 2002. *The English Language*. London: Penguin Books.

Mathieu Dehouck and Marine Delaborde. 2023. Génération automatique de jeux de mots à base de prénoms. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, page 1.

Sophie Jentzsch and Kristian Kersting. 2023. Chatgpt is fun, but it is not funny! humor is still challenging large language models. *Preprint*, arXiv:2306.04563.

Julie-Anne Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Alex Morales and Chengxiang Zhai. 2017. Identifying humor in reviews using background text sources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Copenhagen, Denmark. Association for Computational Linguistics.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. SemEval-2017 task 6: #HashtagWars: Learning a sense of humor. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 49–57, Vancouver, Canada. Association for Computational Linguistics.

Thomas Winters, Vincent Nys, and Daniel De Schreye. 2018. Automatic joke generation: Learning humor from examples. In *Distributed, Ambient and Pervasive Interactions: Technologies and Contexts*, pages 360–377, Cham. Springer International Publishing.

Daniel Zeman, Joakim Nivre, and al. 2022. Universal dependencies 2.11. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.