BU-Pier Team @ CLEF JOKER 2023 Open Task: Slip of the Tongue Generation to Improve Social Interaction with Virtual Agents

Loïc Glemarec[1,2], Fred Charles[2]

[1]Université de Bretagne Occidentale, FR
[2]Bournemouth University, UK

Abstract
We build on the work by [1] which tackled the problem of generating puns in French, by experimenting to generate puns based on the data released for the CLEF 2022 JOKER and inspired by methods for generating English puns with large pre-trained models. We propose, an adaptation of the previous French generation sequence back in English, using an IMSdb modified dataset from [2].

Keywords
Computational Humour, Humour Generation, Wordplay, Computational Creativity

1. Introduction

Laugh and amusement are generally both a resulting effect of *humour*. In everyday life, humour can lead to supporting social interactions [3] between humans by alleviating awkward, uncomfortable, or uneasy feelings. Furthermore, using humour can enhance well-being and help individuals manage anxiety and stress, as well as being effective at reducing distress [4]. According to the benign-violation theory [5], humour only occurs when something seems wrong, unsettling, or threatening, but simultaneously seems okay, acceptable or safe.

Over the years, research into interactive storytelling has produced many prototype systems [6] to engage human users in simulations where virtual agents act out a story [7] influenced by human users' interaction. One of the key characteristics of narratives is tension which has been used as part of implemented systems [8] and it can be found in many dramatic film scripts.

In this paper, we propose a generation system that could be integrated into any other interactive dialogue system. The aim is to generate phonetic-based wordplays that could improve engagement in this kind of interaction. We are giving a "lapsus" format to the puns [1]. Indeed as we are interested in the generation of humour using phonetic composition, we also want to integrate it a posteriori into interactive narration systems. We believe that lapsus is the perfect form of humour, lying somewhere between dialogue and phonetics.

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
loic.glemarec3010@gmail.com (L. Glemarec); fcharles@bournemouth.ac.uk (F. Charles)
https://staffprofiles.bournemouth.ac.uk/display/fcharles (F. Charles)
0000-0001-6207-912X (F. Charles)

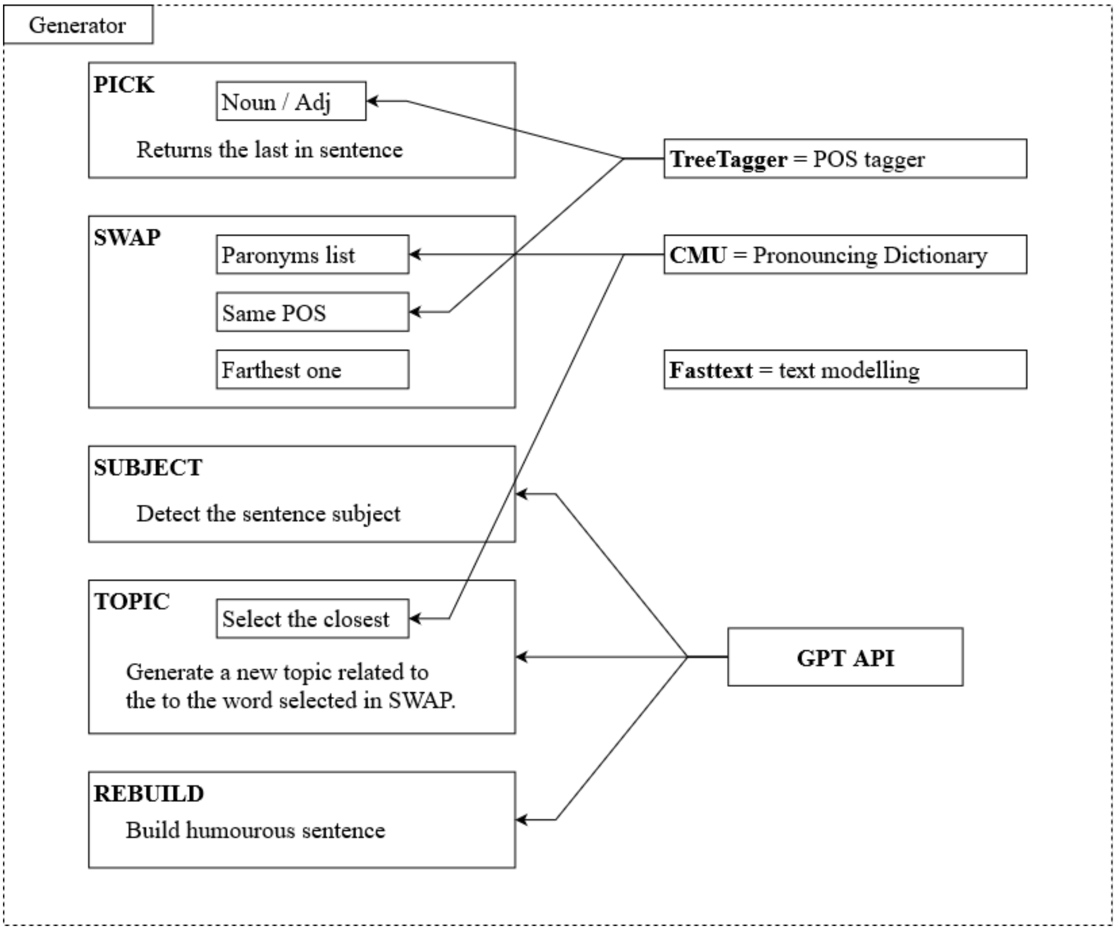

Figure 1: Diagram illustrating which tools are used at which stages - please find in the text below details of the different stages' description (PICK; SWAP; SUBJECT; TOPIC; REBUILD).

2. Method

5-step wordplay generation Our goal is to generate wordplay based on paronyms from a simple sentence without any sense of humour. Wordplay generation is built in 5 distinct steps. To do this, we must first identify the word onto which to apply the wordplay: w_{pick} (PICK). When we have selected a word, we then search the list of all its paronyms and select one: w_{swap} (SWAP). There is also a subject detection step: $w_{subject}$ (SUBJECT). To accentuate the humour effect, we need to change the topic to correspond to w_{swap} so that the context remains consistent: w_{topic} (TOPIC). Finally, once all these generated elements have been brought together, it is possible to rebuild the sentence in the pun format (REBUILD). (see diagram in Figure 1).

PICK For the word selection stage, we start by listing all the adjectives and nouns of the sentence. For this, we use TreeTagger, a part of speech tagger [9]. We then list all the paronyms of the words from the first list. We proceed iteratively and look for paronyms with the same part of speech (i.e. adjectives, nouns,...). If a noun or adjective does not have at least one paronym, it is removed from the list. Finally, when several words are found to have paronyms in the sentence, we select the word closest to the end. This choice is made in order to maximise the surprise and thus the potential for comic effect [10].

SWAP Before proceeding with the exchange of words, we first list all the paronyms by comparing their phonetic form which is provided by a dictionary on word pronunciations[1]. We use the [11] heuristic to determine what are two paronyms, namely two words that are phonetically close. Two words are considered orthographically similar if one word is obtained with a single character deletion, addition, or replacement from the other one. Two words are phonetically similar if their phonetic transcription is orthographically similar according to the above definition. We also added a constraint to improve the selection: the homophone part of speech must be the same as that of the initial word w_{pick}. This allows to retain grammatical coherence. When this condition is met, but there are still several possible homophones, the w_{swap} will be the homophone that is the most semantically distant from w_{pick}. The choice of a semantically distant word permits the selection of the word that will have the most distant context possible. By doing this, the comic effect will be accentuated as the result of increased surprise. To compare the semantic distance of words, we use the English version of *fasttext* [12][2]. This model allows mapping a word to a vector value. The comparison is done by measuring the distance between two word-vectors. The greater the distance, the more semantically distant the two words are. To calculate the distance, we compute the cosine value of the angle formed by the two vectors.

When these operations are successfully completed, we end up with one paronym (w_{swap}) that will provide a grammatically correct substitution and will maximise the potential for humour. The next step is to provide a topic change in the sentence.

SUBJECT Before changing the topic in the sentence, we need to detect the subject ($w_{subject}$). It is achieved through the GPT API[3]. To do so, we provide the model with several examples of sentences while highlighting their subject. This information will be used in the next step and will increase the precision of the generation.

TOPIC The sentence topic change is also operated through the GPT API. As in [1], the topic change is made by changing a word in the sentence. As in the previous step, we provide the model with several examples, then we request the prediction of a new topic. Intending to guide the prediction towards what we are interested in, i.e. consistency between the new paronym and the topic, we provide as information:

- The initial sentence
- The w_{pick} word
- The w_{swap} paronym
- The $w_{subject}$ to change

[1]http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[2]https://fasttext.cc/
[3]https://platform.openai.com/

Table 1
There are a few examples of the 5-step generation.

input	w_{pick}	w_{swap}	$w_{subject}$	w_{topic}	output
Be kind to your dentist because he has feelings too	feelings	filings	dentist	accountant	Be kind to your accountant because he has filings too.
I may look busy, but I'm just confused!	busy	beery	I	The brewer	The brewer may look beery, but he's just confused!
The remedy is worse than the disease	disease	daisies	The remedy	The florist	The florist is worse than the daisies.
I'd love to, but my patent is pending	patent	pattens	I	The shoemaker	The shoemaker would love to, but his pattens is pending.

REBUILD Finally, it is now possible to reconstruct the pun. Again thanks to the GPT API and providing the following information:

- The initial sentence
- The $w_{subject}$ word
- The w_{topic} word

The pun is therefore similar with respect to the initial sentence, but the subject $w_{subject}$ has been changed to w_{topic} to ensure contextual consistency with the paronym w_{swap} of the word w_{pick}.

See Table 1 for samples of generated outputs.

3. Experiment

Our experiment consists in generating *funny* dialogues by integrating puns. To do this, we used dialogues from the IMSdb dataset. We selected scripts from films which systematically include the staging of 2 characters and 2 to 3 utterances per dialogue. This provides a large enough sample of dialogues which are realistic though manageable for our experiment. In this context, a topic shift in the sentences would not make any sense, which is why we have decided not to use the SUBJECT, TOPIC and REBUILD parts of the pipeline for this experiment. Furthermore, the SWAP part will replace w_{pick} with synonyms and antonyms of "foolish", which leads more easily to an incongruous situation. We can thus remain within the context of the dialogue, and we can complete the modified utterances by repairing the slip of the tongue, i.e. by "I mean ..." (See Table 2). m

4. Conclusion

We presented our approach for generating puns based on phonetics following a pipeline in the English language. We moved from a generation using homophones [1] to the use of paronyms, which considerably increases the possibilities. The GPT models seem to perform much better, and we think that the three stages using this model successively can be combined into a single stage, making it possible to go from three to a single request to the API. Furthermore, we aim to integrate this system into the field of narrative interaction.

Table 2
Samples of input sentences and the generated outputs from our end-to-end pipeline.

Inputs	Outputs
- One glass per **man**, sir, Captain's orders. - A mere thimbleful, Corporal, for scientific purposes only.	- One glass per **mad**, sir, Captain's orders. **I mean man !** - A mere thimbleful, Corporal, for scientific purposes only.
- I see you're still conceited. - Could you excuse us, **Dottie**?	- I see you're still conceited. - Could you excuse us, **dotty**? **I mean Dottie !**
- If you were tryin' to **surround**'em, you sure succeeded. - How far's the river from here, Mose?	- If you were tryin' to **sound** 'em, you sure succeeded. **I mean surround !** - How far's the river from here, Mose?
- Excuse me, sorry I'm late. - I'm sure you don't have a good excuse, so I won't force you to come up with a **bad** one. - Thank you, Isaac, that's nice of you. - Sit-down, this first one's for you.	- Excuse me, sorry I'm late. - I'm sure you don't have a good excuse, so I won't force you to come up with a **mad** one. **I mean bad !** - Thank you, Isaac, that's nice of you. - Sit-down, this first one's for you.

TreeTagger is an old tool and we know that other more recent tools have better capabilities for the same task, so we're thinking of using the POS tagging tool from the NLTK library, especially the one that uses Conditionnal Random Fields (CRF). We know that forcing replacement by using a specific lexical field is a double-edged sword: on the one hand it increases the chances of a funny context, but on the other it greatly limits the possibilities. We think it's quite possible to imagine building a custom list, which would group together several lexical fields identified as being potentially effective from the point of view of the funny effect.

5. Online Resources

The sources are available via

- GitLab : https://gitlab.com/loicgle/computational-humour-pun-generation

References

[1] L. Glémarec, A.-G. Bosser, J. Boccou, L. Ermakova, Humorous Wordplay Generation in French, CEUR Workshop Proceedings (2022).

[2] W. Xu, F. Charles, C. Hargood, Generating stylistic and personalized dialogues for virtual agents in narratives, in: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 2023, pp. 737–746.

[3] J. P. Rossing, Prudence and racial humor: Troubling epithets, Critical Studies in Media Communication 31 (2014) 299–313.

[4] N. A. Kuiper, R. A. Martin, Humor and self-concept, HUMOR International Journal of Humor Research (1993).

[5] A. P. McGraw, C. Warren, L. E. Williams, B. Leonard, Too close for comfort, or too far to care? finding humor in distant tragedies and close mishaps, Psychological science 23 (2012) 1215–1223.

[6] M. Cavazza, F. Charles, S. J. Mead, Developing re-usable interactive storytelling technologies, in: R. Jacquart (Ed.), Building the Information Society, Springer US, Boston, MA, 2004, pp. 39–44.

[7] M. Cavazza, F. Charles, S. Mead, Ai-based animation for interactive storytelling, in: Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No.01TH8596), 2001, pp. 113–120. doi:10.1109/CA.2001.982384.

[8] A.-O. Petac, A.-G. Bosser, F. Charles, P. De Loor, M. Cavazza, A pragmatics-based model for narrative dialogue generation, Proceedings of the 11th International Conference on Computational Creativity (2020).

[9] H. Schmid, Probabilistic part-ofispeech tagging using decision trees, in: New methods in language processing, 2013, p. 154.

[10] H. He, N. Peng, P. Liang, Pun generation with surprise, arXiv preprint arXiv:1904.06828 (2019).

[11] A. Valitutti, H. Toivonen, A. Doucet, J. M. Toivanen, "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), 2013, pp. 243–248.

[12] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.