

# Humorous Wordplay Generation in French

Loic Glémarec<sup>1</sup>, Anne-Gwenn Bosser<sup>2</sup>, Julien Boccou<sup>1</sup> and Liana Ermakova<sup>3,4</sup>

<sup>1</sup>Université de Bretagne Occidentale, Brest, France

<sup>2</sup>ENIB, Brest, Lab-STIC CNRS UMR6285

<sup>3</sup>Université de Bretagne Occidentale, HCTI, Brest, France

<sup>4</sup>Maison des sciences de l'homme en Bretagne, Rennes, France

## Abstract

Recent work have tackled the problem of generating puns in English, based on the corpus of English puns from SemEval 2017 Task 7. In this paper, we report on experiments on generating French puns based on the data released for the CLEF 2022 JOKER and inspired by methods for generating English puns with large pretrained models. 50% of generated wellerisms were funny.

## Keywords

Computational Humour, Humour generation, Wordplay, Wellerism, Word embedding, Lexique 3, Large pre-trained models, Few-shot learning, Computational creativity

## 1. Introduction

Humour aims to provoke laughter and provide amusement. The appropriate use of humour can facilitate social interactions [1] as it can reduce awkward, uncomfortable, or uneasy feelings. Humour contributes to higher physical and psychological wellbeing and has shown to be effective to cope with distress [2]. Indeed, according to the benign-violation theory, 'humour only occurs when something seems wrong, unsettling, or threatening, but simultaneously seems okay, acceptable or safe' [3]. Wordplay is a common source of humor because of its subversive and catchy nature. Recent work by [4] have tackled the issue for generating humorous puns in English based on the data provided by [5]. The CLEF Joker Workshop [6, 7] provided a similar dataset for the French language, and allowed us to investigate how well this method could be transposed in French. In the work by [4], the goal is to generate puns in English, relying on paronyms and a modification of the context of the sentence to create surprise, resolving the incongruity that will result in the humorous effect, by applying the pun at the end of sentences. Despite the generality of this principle, most of the published work that we could find on computational humor generation remains primarily for the English language. The work described in [8] makes use of constraints to provide structurally correct and successfully funny wordplay. Their approach, which rely less on statistical linguistic resources than the most recent

---

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

✉ Loic.Glemarec1@etudiant.univ-brest.fr (L. Glémarec); bosser@enib.fr (A. Bosser);


julien.boccou@etudiant.univ-brest.fr (J. Boccou); liana.ermakova@univ-brest.fr (L. Ermakova)

🌐 <https://yamadharmagithub.io/> (L. Glémarec); <https://www.joker-project.com/> (L. Ermakova)

🆔 0000-0002-7598-7474 (L. Ermakova)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

literature, seems appropriate for languages other than English for which these might currently be less performing. It was also an inspiration for our work.

In this paper we show how to generate puns for the French language and that the method works for English as well. We describe a three-steps method: first, we select the word on which to apply wordplay; second, we select one of the most distant homophone semantically, and finally, we look for a novel context consistent with the homophone operated by prediction using a large language model. Using this method, we were able to generate grammatically but also structurally correct wordplay sentences (eg. following the expected pattern or template). Although we have not yet completed a full evaluation of the output, we curated a number of potentially humorous results some of which are provided in this paper.

The incongruity of the expected and given stimuli is also used in wellerisms which exploits the contradiction of figurative and literal meanings. Wellerisms are wordplays that make fun of established clichés and proverbs in a context where they are taken literally [9]. Thus, we also explore the effectiveness of large pre-trained models, such as GPT-3 [10], for wellerism generation.

## 2. Method

### 2.1. 5-step wordplay generation

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

**PICK** For the word selection, we started by listing all the adjectives and nouns of the sentence. For this we use the lexicon Lexique 3 for the French language [11, 12]. We proceeded iteratively and looked for homophones with the same part of speech (adjectives, nouns,...). To ensure grammatical correctness, we limited the selection to words that do not have two possible part of speech for the same spelling. When this is the case, the word is qualified as ambiguous and will therefore not be included in the list of possible targets for wordplay. We check the number of homophones that is information given by Lexique 3. If a noun or adjective does not have at least one homophone, it is also removed from the list. Finally, when several words are found to have homophones 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 comic effect, following the argument in the original paper by [4].

**SWAP** Before proceeding with the exchange, we first listed all the homophones by comparing their phonetic form which is provided as part of Lexique 3. We also added constraints to improve the selection with other key information:

- Flag 1 : The lemmatized form must not be the same as the initial word  $w_{pick}$ .
- Flag 2 : The homophone part of speech must be the same as that of the initial word  $w_{pick}$ . This allows to keep the grammatical coherence.
- Flag 3 : The frequency of occurrence of the homophone must be greater than 2, this information is also given using Lexique 3. Indeed, the two fields *freqlenfilms* and *freqlenlivres* respectively represent the frequency of the lemma according to a corpus of subtitles and the frequency of the lemma according to a corpus of books (both are given per million occurrences). This avoids replacing with a word too little known and therefore creates a feeling of incomprehension.

When these three conditions are 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 one that will have the most distant context possible. By doing this, the comic effect will be accentuated by an increased surprise.

To compare the semantic distance of words, we use the French version of *fasttext* [13]<sup>1</sup>. 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 operation are successfully completed, we end up with one homophone ( $w_{swap}$ ) that will provide a grammatically correct substitution and will maximise the humorous potential. The next step is to provide a topic change in the sentence.

**SUBJECT** Before topic change in the sentence, we need to detect the subject ( $w_{subject}$ ). It is achieved through Jurassic<sup>2</sup> [14], a Large Language Model. To do so, we provide the model with several examples (see in A) of sentences while highlighting their subject. This information will be use in next step and will permit a better generation precision.

**TOPIC** The sentence topic change is also operated through Jurassic. As in the research we based our proposal on, the topic change is made by changing a word in the sentence. As in previous step, we provide the model with several examples (B), which were constructed using the dataset from the CLEF 2022 Joker Workshop [6] then we request the prediction of a new topic for the setup provided by the previous steps for some other example from the test dataset. Intending to guide the prediction toward what we are interested in, i.e. consistency between the new homophone and the topic, we give as information:

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

We asked for the generation of 15 predictions. This is followed by the removal of duplicates before selecting the subject ( $w_{topic}$ ) most semantically close to  $w_{swap}$ .

---

<sup>1</sup><https://fasttext.cc/>

<sup>2</sup><https://studio.ai21.com/docs/jurassic1-language-models/>

**REBUILD** Finally, it is now possible to reconstruct the pun. Again thanks to Jurassic and providing the following information (C):

- 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 homophone  $w_{swap}$  of the word  $w_{pick}$ .

## 2.2. Wellerism generation with large pre-trained models

Wellerisms are wordplay that make use of catchphrases, phrases or expressions recognized by their repeated utterance. Wellerisms are a common type of wordplay with recognizable conventional form which helps to prepare a joke. We generated the following types of wellerisms that were:

- **Question-Answer.** This type of wellerisms refers to bipartite jokes with the form of a question followed by an answer.

**Example 2.1.** Qu'est-ce que l'étudiant dit à la calculatrice? Tu comptes beaucoup pour moi.

- **Old soldiers never die** wellerisms are transformations of the catchphrase, with the full version being *Old soldiers never die, they simply fade away*.

**Example 2.2.** Les vieux électriciens ne meurent pas, ils 100 volts.

- **Tom swifty** are wellerisms with a phrase in which a quoted sentence is linked by a pun to the manner in which it is attributed. The standard form is for the quoted sentence to be first, followed by the description of the act of speaking of the conventional speaker *Tom*

**Example 2.3.** "J'ai commencé à lire Voltaire", avoua Tom d'un ton candide.

To generate these types of wellerisms, we used prompt-tuning of large-pretrained models, namely GPT-3 [10]. Discrete prompt-tuning is a widely-used technique to condition frozen language models to perform specific downstream tasks [15]. We considered the generation of each type of these wellerisms as an individual task.

The prompts were generated automatically based on the data in French constructed at the JOKER workshop [6, 7]. We applied regular expressions to extract *Question-Answer*, *Old soldiers never die* and *Tom swifty* wellerisms from the corpus. We generated a training prompt for each category by randomly selecting small training set from the corresponding subcorpus, i.e. we used three distinct training prompts in total. The same training prompt was applied for all generations. As all these wellerisms are bipartite, we split wordplay into two parts and we used the first part of each wordplay for the generation.

**Table 1**

Statistics of data used for 5-step generation

Step	# requests number	# train instances	# output
TOPIC	1	5	15
SUBJECT	1	6	1
REBUILD	1	5	1

**Table 2**

Statistics of data used for wellerism generation

Wellerism category	# train instances	# test instances	# total in corpus
Question-Answer	20	50	392
Old soldiers never die	10	40	272
Tom swifty	20	50	503

### 3. Evaluation framework

#### 3.1. Data description

Our data is twofold, containing human and machine translations of the SemEval-2017 corpus of English puns [5] into French [7].

The 5-step wordplay generation aims to transform a non-humorous text into wordplay. Thus, the source corpus should without wordplay but with a potential to do it. With this consideration in mind, we used machine translations generated by the participants of *JOKER Task 3: Pun translation from English into French* [7]. We used only the machine translations that were annotated not to contain wordplay. The initial non-wordplay corpus consisted of 6 780 texts. As machine translations of the same source text might be quite similar, we dropped entries with duplicated identifiers of source puns in English and, thus, kept only one machine translation per English pun. Then, we filtered out texts for which we could not find homophones in *Lexique-3*.

The French wordplay subcorpus used for wellerism generation is a subset of human translations of the SemEval-2017 English puns [5] produced during at the JOKER translation contest [7]. We used a small subset for the training part of the prompt and 40-50 examples for generation. Although it is impossible to use a direct comparison of test data with generation for the evaluation due to multiple ways to play on words, the use of joke parts guarantee the possibility of wordplay. The details of the data statistics is given in Table 2. Although, the corpus contains 272 *Old soldiers never die* wellerisms, we found only 50 distinct subject. Thus, we used 10 subjects for training and 40 subjects for test wellerism generation.

#### 3.2. Annotation and evaluation metrics

A master student in translation, French native speaker, manually annotated the produced generation according to the following binary categories:

- wordplay presence;

**Table 3**  
Results of generation

Category	Wordplay	Non-sens	Truncated	Syntax problem	Lexical problem
Question-Answer	8 (15%)	9	2	2	5
Tom swifty	15 (30%)	0	0	11	8
Old soldiers never die	26 (65%)	6	0	1	3
5-step	7(8%)	49	0	2	9

- non-sens;
- truncated text;
- syntax problem;
- lexical problem.

We applied the Likert scale [16] to evaluate joke hilariousness. We applied the scale from 0 to 5 referring to humorless and the funniest texts respectively. The annotator was also asked to provide free comment on jokes.

We report absolute values as well as the percentage of wordplay in generated texts.

## 4. Results

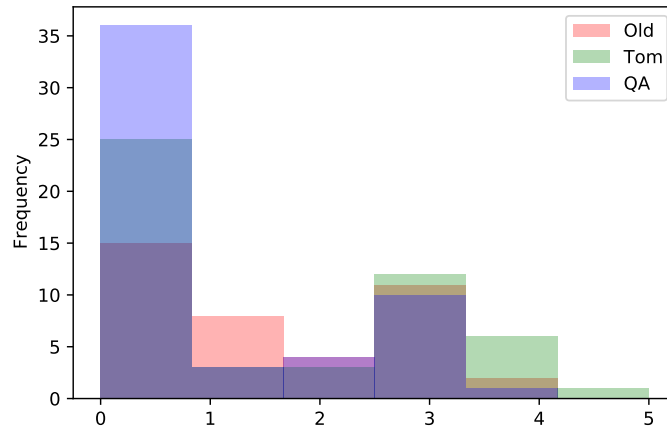
Table 3 shows the results of generation. The generation for the category *Old soldiers never die* was the most successful, with 65% wordplay produced. Notice that this category is the most homogeneous, as the beginning varies only in the subject. We observe a significant drop (twice lower wordplay rate) for the *Tom swifty* jokes which has a more heterogeneous form and the lowest results were demonstrated for the *Question-Answer* type as their form is the less strict.

Figure 1 presents the histogram of the hilariousness scores of generated wellerisms. Almost 50% of generated wellerisms were judged funny by the French native speaker annotator, i.e. they were attributed a hilariousness score  $\geq 1$ . The most successful jokes were *Tom swifty* while the vast majority of *Question-Answering* were judged non-humorous. *Question-Answering* was the most heterogeneous category.

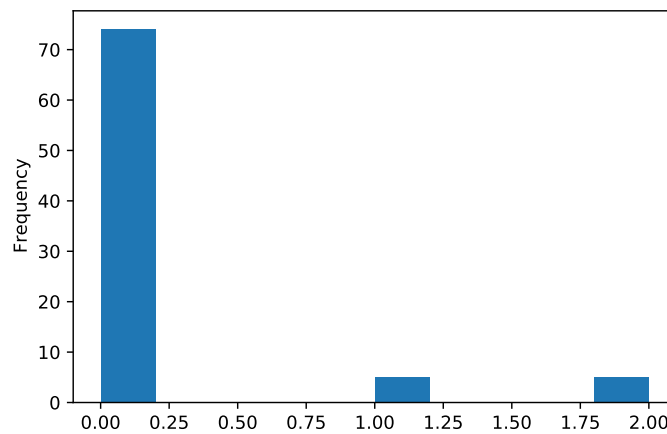
The statistics on free category for generated wellerisms is given in Table 4. As it is evident from the table, the *Old soldiers never die* wellerisms are considered to be euphemisms in 90% of cases.

Deadpans occur in *Question-Answer* jokes. Deadpan, also called dry or dry-wit humor, is a form of comedic delivery with the deliberate display of emotional neutrality contrasting with the ridiculousness or absurdity of the subject matter [17]. The delivery is meant to be blunt, ironic, or unintentional. We do not observe deadpans in *Tom swifty* nor *Old soldiers never die* wellerisms as they do not have interaction with interlocutor nor environment.

The results generated using the 5-steps method are grammatically correct and structurally correspond to our expectations: the PICK phase keeps the part of speech of the word, and



**Figure 1:** Histogram of the hilariousness scores of generated wellerisms



**Figure 2:** Histogram of the hilariousness scores of 5-step generation

discards ambiguous words on this subject. This limits the number of grammatically false results, which could harm the humorous effect. The homophonic criteria being easy to use with a phonetic lexicon and providing obvious wordplay. These may limit the potential generation when compared to computing paronyms.

In D, you can find examples of puns we created. Each box contains an original sentence, as well as the built pun. These examples are also used for prompts for jurassic. Then, in E, you can find the test sentences that fulfill constraints. Sentences in F are all tagged as ambiguous,

**Table 4**

Free category statistics for generated wellerisms

Category	Blunt	Absurdism	Jokes for Kids	Euphemism	Dark humor	Deadpan	One liner	Poetic
QA	.	3	3	.	2	5	1	.
Tom	2	4	.	.	.	.	.	.
Old	2	.	.	36	.	.	2	3
5-step	.	1	.	.	.	.	.	.

even if they seem to be structurally encouraging, each one contains words that have known homophones adequately placed in the sentences.

Here are selected results:

- **Comte / Compte** (*Count/Account*)

*Original sentence* : Nous avons beaucoup voyagé, mais grace au comte.

*(We have traveled a lot, but thanks to the count.)*

*Generated*: La monnaie a beaucoup voyagé, mais grace au compte.

*(The currency has traveled a lot, but thanks to the account.)*

- **Encre / Ancre** (*Anchor/Ink*)

*Original sentence* : Le marin au milieu de rien, a jeté son ancre.

*(The sailor in the middle of nothing, dropped his anchor.)*

*Generated* : Le poète au milieu de rien, a jeté son encre.

*(The poet in the middle of nothing, threw his ink.)*

- **Mère / Mer** (*Mother/Sea*)

*Original sentence* : Le fils a dit au revoir à sa mère.

*(The son said goodbye to his mother.)*

*Generated* : Le morse a dit au revoir à sa mer.

*(The walrus said goodbye to its sea.)*

- **Seau / Sot** (*Bucket/Fool*)

*Original sentence* : L'ouvrier a fait tomber un seau.

*(The worker dropped a bucket.)*

*Generated* : Un casse-tête a fait tomber un sot.

*(A puzzle knocked down a fool.)*

The current implementation is a proof of concept, and the generativity of the solution could be improved in several ways.

The PICK phase provides coherence, but restricted to certain part of speech and forms of word for simplicity. We will later improve the range of homophones that can be used. We also plan to provide more variety during the SWAP phase, by expanding the puns to include paronyms instead of the restricted case of homophones.



This led us to wonder about what criteria to use for deciding when and how paronyms are perceived and understood by humans. Whilst the answer to this question is likely to be context specific, skill specific (such as in *Contrepèterie*/spoonerism identification) and depend on the media used to communicate the pun (script, voice), we can already consider several criteria to take into account in identifying which words may be likened by humans in understanding puns, for instance: phonetic transcription closeness (Hamming distance), similar number of syllables, similar structure in terms of vowels and consonants, rhymes.

We plan to investigate whether and to which extent various types of punning criteria allow us to generate more varied and less obvious puns, which may render them more satisfying depending on the human audiences. Finally, the TOPIC step of our method was merely a first investigation of Large Language Models and can certainly be improved, especially in terms of contextual relevance between the homophone and the new topic. We plan to look for more effective prompts, to influence Jurassic's prediction.

## 5. Conclusion

We have presented the results of first investigations for generating humorous puns in French using large pre-trained language models. The first method is based on work previously done for the English language, with some adaptation to account for linguistic resources available for the French language. Our method proceeds in five steps which will transform a sentence: the PICK step quickly selects the target word for the pun, the SWAP step defines whether the word is replaceable by a homophone, the SUBJECT step retrieves the input sentence subject, the TOPIC step which, thanks to the Jurassic model, predicts a new, contextually coherent topic for the original sentence, and the REBUILD step build the final pun with Jurassic too. We have presented a few encouraging results. We plan to investigate this topic further: in particular we would like to work on extending puns to include a variety of structures and heuristics that humans use to recognize paronyms in punning, and try out different models. We also tried out the generation of wellerism using the GPT-3 Model, with prompt-tuning using puns sharing similar templates, with promising results. 50% of generated wellerisms were funny.

## 6. Online Resources

The sources for the generation of humorous puns in French are available via

- GitLab : <https://gitlab.com/loicgle/computational-humor-pun-generation>,

## References

- [1] J. P. Rossing, Prudence and racial humor: Troubling epithets, *Critical Studies in Media Communication* 31 (2014) 299–313.
- [2] N. A. Kuiper, R. A. Martin, *Humor and self-concept*, Walter de Gruyter, Berlin/New York Berlin, New York, 1993.

- [3] 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.
- [4] H. He, N. Peng, P. Liang, 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)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1734–1744. URL: <https://aclanthology.org/N19-1172>. doi:10.18653/v1/N19-1172.
- [5] T. Miller, C. Hempelmann, I. Gurevych, SemEval-2017 task 7: Detection and interpretation of English puns, in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 58–68. URL: <https://aclanthology.org/S17-2005>. doi:10.18653/v1/S17-2005.
- [6] L. Ermakova, T. Miller, O. Puchalski, F. Regattin, E. Mathurin, S. Araújo, A.-G. Bosser, C. Borg, M. Bokiniec, G. L. Corre, B. Jeanjean, R. Hannachi, G. Mallia, G. Matas, M. Saki, CLEF Workshop JOKER: Automatic Wordplay and Humour Translation, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, V. Setty (Eds.), *Advances in Information Retrieval, Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2022, pp. 355–363. doi:10.1007/978-3-030-99739-7\_45.
- [7] L. Ermakova, T. Miller, F. Regattin, A.-G. Bosser, E. Mathurin, G. L. Corre, S. Araújo, J. Boccou, A. Digue, A. Damoy, B. Jeanjean, Overview of JOKER@CLEF 2022: Automatic Wordplay and Humour Translation workshop, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022)* 13390 (2022).
- [8] 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)*, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 243–248. URL: <https://aclanthology.org/P13-2044>.
- [9] L. Lundin, Wellerness, WELLERISMS AND TOM SWIFTIES. ORLANDO: SLEUTHSAYERS (2011).
- [10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>.
- [11] B. New, C. Pallier, M. Brysbaert, L. Ferrand, Lexique 2 : A new French lexical database, *Behavior Research Methods, Instruments, & Computers* 36 (2004) 516–524. URL: <https://doi.org/10.3758/BF03195598>. doi:10.3758/BF03195598.
- [12] C. Pallier, B. New, J. Bourgin, Openlexicon, GitHub repository, 2019. URL: <https://github.com/chrplr/openlexicon>.
- [13] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword

- information, Transactions of the Association for Computational Linguistics 5 (2016). doi:10.1162/tacl\_a\_00051.
- [14] O. Lieber, O. Sharir, B. Lenz, Y. Shoham, JURASSIC-1: TECHNICAL DETAILS AND EVALUATION, White Paper, AI21 Labs, 2021. URL: [https://uploads-ssl.webflow.com/60fd4503684b466578c0d307/61138924626a6981ee09caf6\\_jurassic\\_tech\\_paper.pdf](https://uploads-ssl.webflow.com/60fd4503684b466578c0d307/61138924626a6981ee09caf6_jurassic_tech_paper.pdf).
- [15] B. Lester, R. Al-Rfou, N. Constant, The Power of Scale for Parameter-Efficient Prompt Tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. URL: <https://aclanthology.org/2021.emnlp-main.243>. doi:10.18653/v1/2021.emnlp-main.243.
- [16] R. Likert, A technique for the measurement of attitudes, Archives of Psychology 22 140 (1932) 55–55.
- [17] M. A. Rishel, Writing Humor: creativity and the comic mind, Wayne State University Press, 2002.

## Appendices

### Appendix A SUBJECT Jurassic Prompt

```
Repérer le sujet de la phrase
--
Phrase: l'homme est tombé, il a cassé un verre
Sujet: homme
--
Phrase: il a dû se confiner, il a pris du poids
Sujet: il
--
Phrase: tu devrais faire attention à ce que tu manges, ce n'est pas très sain
Sujet: tu
--
Phrase: le français est la dixième langue maternelle la plus parlée au monde
Sujet: français
--
Phrase: après la fête, j'ai passé le balai
Sujet: je
--
Phrase: pour l'anniversaire du voisin, il y avait un gâteau aux amandes
Sujet: voisin
--
```

## Appendix B TOPIC Jurassic Prompt

Changez le contexte de la phrase selon l'homonyme remplaçant.

--

phrase: L'homme est tombé, il a cassé un verre.

mot à remplacer: verre

homonyme: vers

nouveau contexte de l'homonyme remplaçant: Le poète

--

phrase: Il a du se confiné, il a pris du poids.

mot à remplacer: poids

homonyme: pois

nouveau contexte de l'homonyme remplaçant: Le maraicher

--

phrase: Tu devrais faire attention à ce que tu mange, ce n'est pas très sain.

mot à remplacer: sain

homonyme: saint

nouveau contexte de l'homonyme remplaçant: le pape

--

phrase: Après la fête, il a passé le balai.

mot à remplacer: balai

homonyme: ballet

nouveau contexte de l'homonyme remplaçant: le danseur

--

phrase: Pour l'anniversaire du voisin, il y avait un gâteau aux amandes.

mot à remplacer: amandes

homonyme: amendes

nouveau contexte de l'homonyme remplaçant: policier

--

## Appendix C REBUILD Jurassic Prompt

Changer le sujet de la phrase initiale avec le nouveau sujet pour créer une nouvelle phrase.

--

Phrase: L'homme est tombé, il a cassé un verre.

Sujet: l'homme

Nouveau sujet: le poète

Nouvelle phrase: Le poète est tombé, il a cassé un verre.

--

Phrase: Il a dû se confiner, il a pris du poids.

Sujet: il

Nouveau sujet: le maraicher

Nouvelle phrase: Le maraicher a dû se confiner, il a pris du poids.

--

Phrase: Tu devrais faire attention à ce que tu manges, ce n'est pas très sain.

Sujet: tu

Nouveau sujet: le prêtre

Nouvelle phrase: le prêtre devrait faire attention à ce qu'il mange, ce n'est pas très sain.

--

Phrase: Après la fête, il a passé le balai.

Sujet: il

Nouveau sujet: le danseur

Nouvelle phrase: Après la fête, le danseur a passé le balai.

--

Phrase: Pour l'anniversaire du voisin, il y avait un gâteau aux amandes.

Sujet: voisin

Nouveau sujet: le policier

Nouvelle phrase: Pour l'anniversaire du policier, il y avait un gâteau aux amandes.

--

## Appendix D Examples of before and after generation

L'**homme** est tombé, il a cassé un **verre**.  
*Le poète* est tombé, il a cassé un *vers*.

Il a dû se confiner, il a pris du **poids**.  
*Le maraicher* a dû se confiner, il a pris du *pois*.

Tu devrais faire attention à ce que tu manges, ce n'est pas très **sain**.  
*Le pape* devrait faire attention à ce que qu'il mange, ce n'est pas très *saint*

Après la fête, **il** a passé le **balai**.  
Après la fête, *le danseur* a passé le *ballet*.

Pour l'anniversaire du **voisin**, il y avait un gâteau aux **amandes**.  
Pour l'anniversaire du *policier*, il y avait un gâteau aux *amendes*.

## Appendix E French Test Sentences

La comtesse viendra, mais accompagné du **comte**. - *compte*

Le marin frustré, a jeté son **ancre**. - *encre*

Le fils à dit au revoir a sa **mère**. - *mer*

L'ouvrier a fait tomber un **seau**. - *sot*

## Appendix F Ambiguous test sentences

Les employés sont heureux, ils vont recevoir leur **paie**.  
Nous avions faim, nous avons mangé des **pâtes**.  
Quand je stresse, j'ai des **tics**.  
J'aime ce genre de musique, surtout avec ce genre de **ton**.  
Vous avez beau essayer, c'est en **vain**.  
L'élève sur le long terme, dépassera toujours le **maître**.  
Je suis fatigué, et j'en ai marre de l'**être**.  
Il est parti pour bivouaquer, et a emporté ma **tente**.