Dutch Humor Detection
by Generating Negative Examples

Thomas Winters[0000−0001−7494−2453] and Pieter Delobelle[0000−0001−5911−5310]

Dept. of Computer Science; Leuven.AI
KU Leuven, Belgium
`firstname.lastname@kuleuven.be`

Abstract. Detecting if a text is humorous is a hard task to do computationally, as it usually requires linguistic and common sense insights. In machine learning, humor detection is usually modeled as a binary classification task, trained to predict if the given text is a joke or another type of text. Rather than using completely different non-humorous texts, we propose using text generation algorithms for imitating the original joke dataset to increase the difficulty for the learning algorithm. We constructed several different joke and non-joke datasets to test the humor detection abilities of different language technologies. In particular, we compare the humor detection capabilities of classic neural network approaches with the state-of-the-art Dutch language model RobBERT. In doing so, we create and compare the first Dutch humor detection systems. We found that while other models perform well when the non-jokes came from completely different domains, RobBERT was the only one that was able to distinguish jokes from generated negative examples. This performance illustrates the usefulness of using text generation to create negative datasets for humor recognition, and also shows that transformer models are a large step forward in humor detection.

Keywords: Computational Humor · Humor Detection · RobBERT · BERT model

1 Introduction

Humor is an intrinsically human trait. All human cultures have created some form of humorous artifacts for making others laugh [5]. Most humor theories also define humor in function of the reaction of the perceiving humans to humorous artifacts. For example, according to the popular incongruity-resolution theory, we laugh because our mind discovers that an initial mental image of a particular text is incorrect, and that this text has a second, latent interpretation that only becomes apparent when the punchline is heard [25,24]. To determine that something is a joke, the listener thus has to mentally represent the set-up, followed by detecting an incongruity caused by hearing the punchline, and resolve this by inhibiting the first, non-humorous interpretation and understanding the second interpretation [8,14]. Such humor definitions thus tie humor to the

abilities and limitations of the human mind: if the joke is too easy or too hard for our brain, one of the mental images might not get established, and lead to the joke not being perceived as humorous. As such, making computers truly and completely recognize and understand a joke would not only require the computer to understand and notice the two possible interpretations, but also that a human would perceive these as two distinct interpretations. Since the field of artificial intelligence is currently nowhere near such mental image processing capacity, truly computationally understanding arbitrary jokes seems far off.

While truly understanding jokes in a computational way is a challenging natural language processing task, there have been several studies that researched and developed humor detection systems [26,20,17,6,2,29,1]. Such systems usually model humor detection as a binary classification task where the system predicts if the given text is a joke or not. The non-jokes often come from completely different datasets, such as news and proverbs [20,36,6,2,7,1]. In this paper, we create the non-joke dataset by using text generation algorithms designed to mimic the original joke dataset by only using words that are used in the joke corpus [31]. This dataset thus substantially increases the difficulty of humor detection, especially for algorithms that use word-based features, given that coherence plays a more important role in distinguishing the two. We use the recent RobBERT model to test if its linguistic abilities allow it to also tackle the difficult challenge of humor detection, especially on our new type of dataset. As far as the authors are aware, this paper also introduces the first Dutch humor detection systems.

2   Background

2.1   Neural Language Models

Neural networks perform incredibly well when dealing with a fixed number of features. When dealing with sequences of varying lengths, recurrent connections to previous states can be added to the network, as done in recurrent neural networks (RNN). Long short-term memory (LSTM) networks are a variety of RNN that add several gates for accessing and forgetting previously seen information. This way, a sequence can be represented by a fixed-length feature vector by using the last hidden states of multiple LSTM cells [15]. Alternatively, if a maximum sequence length is known, the input size of a neural network could be set to this maximum, and e.g. allow for using a convolutional neural network (CNN).

Entering text into a recurrent neural network is usually done by processing the text as a sequence of words or tokens, each represented by a single vector from pre-trained embeddings containing semantic information [21]. These vectors are obtained from large corpora in the target language, where the context of a token is predicted e.g. using Bag-of-Words (BOW) [21].

BERT The BERT model is a powerful language model that improved many state-of-the-art performances on NLP tasks [11]. It is built using a transformer encoder stack consisting of self-attention heads to create a bidirectional language

model [28]. These attention mechanisms allow BERT to distinguish different meanings for particular words based on the context by using contextualized embeddings. For example, even though the word *"stick"* could be both a noun as well as a verb, normal word embeddings assign the same vector to both meanings.

BERT is trained in a self-supervised way by predicting missing words in sentences, and predicting if two randomly chosen sentences are subsequent or not. After this pre-training phase, additional heads can be fine-tuned on particular datasets to classify full sentences, or to classify every token of a sentence. The model exhibits large quantities of linguistic knowledge (e.g. for resolving coreferences, POS tagging, sentiment analysis) and achieved state-of-the-art performance on many different language tasks. This model later got critically re-evaluated and improved in the RoBERTa model, which uses a revised training regime [19].

RobBERT RobBERT [10] is a recent Dutch language model trained using the RoBERTa regime [19] on the Dutch OSCAR corpus section [22]. Like RoBERTa, RobBERT also outperforms other Dutch language models in a wide range of complex NLP tasks e.g. coreference resolution and sentiment analysis [10].

2.2   Humor Detection

Humor is not an easy feat for computational models. True humor understanding would need large quantities of linguistic knowledge and common sense about the world to know that an initial interpretation is being revealed to be incompatible with the second, hidden interpretation fitting the whole joke rather than only the premise. Many humor detection systems use hand-crafted (often word-based) features to distinguish jokes from non-jokes [26,20,17,2]. Such word-based features perform well when the non-joke dataset is using completely different words than the joke dataset. From humor theory, we know that the order of words matter, since stating the punchline before the setup would only cause the second interpretation of the joke to be discovered, making the joke lose its humorous aspect [24]. Since word-based humor detectors often fail to capture such temporal differences, more contextual-aware language models are required to capture the true differences between jokes and non-jokes.

Using a large pre-trained model like the recent BERT-like models is thus an interesting fit for the humor detection task. One possible downside is that these models are not well suited for grasping complex wordplay, as their tokens are unaware of relative morphological similarities, due to the models being unaware of the letters of the tokens [3]. Nevertheless, BERT-like models have performed well on English humor recognition datasets [29,1].

Recently, several parallel English satirical headline corpora have been released for detecting humor, which might help capture subtle textual differences that create or remove humor [30,16]. Lower resource languages however usually do not have access to such annotated parallel corpora for niche tasks like humor detection. While there has been some Dutch computational humor research [31,32,33], there has not been any published research about Dutch humor detection, nor are there any public Dutch humor detection data sets available.

2.3   Text Generation for Imitation

There are many types of text generation algorithms. The most popular type of text generation algorithms use statistical models to iteratively predict the next word given previous words, e.g. n-gram based Markov models or GPT-2 and GPT-3 [23,4]. These algorithms usually generate locally coherent text [31]. A common downside is that the output tends to have globally different structures than the training data (e.g. much longer or shorter), or even break (possibly latent) templates of the dataset [34].

Dynamic Templates Templates can be seen as texts with holes, which are later filled, and thus enforce a global textual structure. One approach for learning these is the dynamic template algorithm (DT), which is designed to replace context words with other, grammatically similar words [31]. It achieves this by analyzing the part-of-speech (POS) tags in the dynamic template text and replaces these words with context words with the same POS tags. It prioritizes low unigram-frequency words, as these are usually key words determining the context of the text. This way, the dynamic template algorithm generates a large variety of more nonsensical versions of given texts, using only words from the corpus.

3   Data

3.1   Collecting Datasets

We collected a Dutch joke dataset by combining the jokes found on Kidsweek[1], DeBesteMoppen[2] and LachJeKrom[3]. This resulted in a dataset of **3235 jokes**.

For the non-joke datasets, we first collected several datasets inspired by the type of datasets used in English humor detection, namely proverbs and news [20,36,6,2,7,29,1]. The proverbs dataset originates from the Dutch proverbs Wikipedia page and contains **1887 proverbs**. The news dataset are **3235 headlines** uniformly sampled from the 100K Dutch news headlines dataset [37].

3.2   Negative Generation: Generating Non-Jokes from Jokes

Since news and proverbs use completely different words and structures, there is a need for a new type of challenging dataset for humor recognition that uses non-jokes that are close to jokes. Given the fragile nature of a joke, changing several important words usually turn the joke into a non-humorous text. We propose a new type of dataset for humor detection by generating negative examples by automatically imitating the joke dataset. The dynamic template algorithm is a right fit for this, as it will not change the global structure like Markov models might do and is less prone to plagiarising large parts of the training corpus [31].

---

[1] https://www.kidsweek.nl/moppen
[2] https://www.debestemoppen.nl/
[3] https://www.lachjekrom.com/

The DT algorithm creates absurd, but globally similar texts, by grammatically similar words into another joke. For example, the joke *"Wat is groen en plakt aan de muur? Kermit de sticker!"*[4] was turned into the non-joke *"Wat is groen en telefoneert aan de muur? Kermit de spin!"*[5].

We chose the same parametrisation used in the original paper (see Appendix A) [31]. The resulting non-jokes thus only use words from the jokes dataset, with comparable frequencies, and still have similar grammatical structures, albeit nonsensical content. This way, language classifiers that just learn which words are more common in jokes (e.g. *"oen"*, *"Jantje"*, *"blond"*...) will be at a disadvantage compared to models that have better insight in the semantic coherence of a joke. Another advantage of this method for parallel corpus creation is that it is easily extensible to other lower resource languages.

4 Evaluation

We devised two types of learning tasks for detecting humor in these new datasets. The first is the classic humor detection task with binary labels representing joke and non-joke. The second is a pairwise humor detection task, where given a joke and a non-joke, the algorithm needs to detect which of the two is a joke.

4.1 Models

We compare four different models[6], namely a Naive Bayes classifier with the TF-IDF of 3000 (1,3)-grams as features, an LSTM with Dutch word embeddings [27], a CNN with two convolutional layers and max pooling on Dutch word embeddings [18], and RobBERT [10]. The use of LSTMs and CNNs allows us to compare the RobBERT model with the previous generation of neural language models.

4.2 Classification Experiment

In this binary classification experiment, the models classifies a given text as a joke or a non-joke. We compared three different datasets, comparing jokes with news, with proverbs, and with generated jokes using the dynamic template algorithm. We performed a random hyperparameter search with 10 runs for the LSTM, CNN, and RobBERT. The full search space and other hyperparameters are listed in the Appendix in Table 2 and Table 3. In addition, we use these *random* hyperparameter trials to estimate the maximum validation accuracy [12]. This allows us to compare performance without it being caused by a computational budget favoring one model. Figure 1 shows these estimates for the validation accuracy for all three datasets. For both the news (Figure 1a) and proverbs (Figure 1b), both the

---

[4] *"What's green and adheres to the wall? Kermit the Sticker"*, pun on *"kikker"* (*"frog"*)

[5] *"What's green and telephones on the wall? Kermit the Spider"*

[6] The code, models, data collectors and demo are available on https://github.com/twinters/dutch-humor-detection.
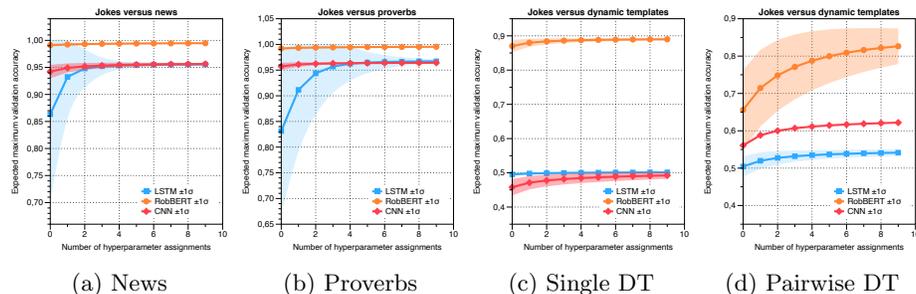
Fig. 1: Estimated maximum validation accuracy [12] in function of the number of hyperparameter trials for the LSTM, CNN, and RobBERT models.

CNN and the LSTM-based models perform a couple of percentage points below the RobBERT model. More notably, the RobBERT model consistently achieves a validation accuracy around 99%, whilst the LSTM has a higher variance than both the CNN model and RobBERT. This indicates that the LSTM-based model is less robust to suboptimal hyperparameter assignment.

From these randomized trials, we select the best-performing model using the validation accuracy and evaluate on the held-out test set. The results are presented in Table 1. The baseline, Naive Bayes, performs quite poorly, with the results being no better than random on all three datasets. This is surprising, given that this method has been used successfully for humor detection in English for similar types of datasets, albeit often using handcrafted features instead of words [20,17,2]. This shows that a classifier using only token features is insufficient for all three Dutch humor datasets. The LSTM and CNN models recognise about 94% for the simple datasets However, they both fail at distinguishing between jokes and non-jokes generated with dynamic templates. This indicates that despite using Dutch word embeddings, these models likely still relies on vocabulary differences or the small lengths news and proverbs tends to have.

Finetuning RobBERT gives us a testing accuracy of 98.8% and 99.6% on *news* and *proverbs*, respectively, and 89.2% on the more challenging task with dynamic templates. This shows that our newly created dataset is indeed more challenging than using non-jokes from completely different domains. Interestingly, RobBERT's false positives contain many jokes with only limited replaced words, or that still made semantically coherent sense, e.g. the joke *"Hoe heet de broer van Bill Mars? Bill Twix!"*[7], only had one replacement (*"Bruno" to "Bill'*), retaining some of the joke's humor. RobBERT's higher performance than the other neural networks also illustrates the advantage of pre-trained language models for detecting semantic coherence in jokes, or at least distinguishing it from semantically incoherent non-jokes generated by the DT algorithm, either of which are useful properties. To get more grasp on this, we classified all elements of the news and the proverbs dataset using the finetuned RobBERT model for the

---

[7] *"What's the name of Bill Mars' brother? Bill Twix!"*

Table 1: Classification results for the held-out test set on three datasets versus the Jokes dataset. For the accuracy, we additionally report the 95% CI.

| | News | | Proverbs | | Dynamic template | | | |
| | | | | | Single | | Pairwise | |
Model	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]	ACC [%]	F_1 [%]
Naive Bayes	51.0 ± 3.1	49.3	60.2 ± 3.5	50.5	49.9 ± 3.1	49.9	-	-
LSTM	94.0 ± 1.5	94.0	94.4 ± 1.6	94.1	46.8 ± 3.1	35.2	47.9 ± 4.4	32.4
CNN	93.6 ± 1.5	93.6	94.1 ± 1.7	93.6	47.4 ± 3.1	46.3	58.6 ± 4.4	58.5
RobBERT	$\mathbf{98.8 \pm 0.7}$	**98.8**	$\mathbf{99.6 \pm 0.4}$	**99.6**	$\mathbf{89.2 \pm 1.9}$	**89.1**	$\mathbf{82.51 \pm 3.4}$	**82.5**

jokes versus dynamic template single setting. This input data was thus completely out-of-domain for this model. We found that 93.23% of the news and 73% of the proverbs were labeled as a joke by this model, indicating that at least for such relatively short strings, the DT model might rely on topical or semantic coherence to recognise humor.

4.3 Pairwise Classification Experiment

We additionally perform an experiment where the joke and their non-joke counterpart, generated by the DT algorithm, are directly compared in a pairwise fashion. The model thus has to recognise which one is more humorous than the other, opening the way for humor preference learning algorithms [13].

We evaluated an LSTM model with two separate recurrent layers with trainable Dutch word embeddings that are concatenated before a fully connected layer, which was also used for argument classification [9]. We evaluated a CNN model following a similar approach, with the same base architecture as in the previous experiment. For RobBERT, we are using the same setup and hyperparameters, and feed the model both texts simultaneously, separated by the separator token.

In Table 1, we can see that LSTMs are still unable to distinguish jokes from generated non-jokes, and CNNs only seeing a small performance boost in the pairwise case over the single case, illustrating the advantage of using such a challenging dataset. RobBERT on the other hand is performing reasonably well but surprisingly loses some accuracy compared to the single classification case. This is likely due to relatively more of the jokes being truncated to fit its input size limit, given that two texts are now fitted into the same input space.

5   Future Work

One way to improve the humor detection performance could be finding better ways of generating joke-like non-jokes, thus further increasing the difficulty of the dataset. The DT algorithm is prone to occasional grammatical errors, which the models might pick up and use to just recognize grammatical errors, rather than recognize jokes.

These new humor detection algorithms also pave way for new humor generators, e.g. using a generate-and-test approach [35]. RobBERT could even fulfill two roles in such a generator, e.g. using a genetic algorithm that uses a pairwise joke detection head as tournament selection, and the word masking head to mutate the genomes. Such a generator could also be useful in a collaborative setting where the humor comparator suggests better ways of phrasing a joke by subtly changing it e.g. by rearranging a potential punchline word to occur later.

6   Conclusion

We created three datasets for humor detection specifically for Dutch and proposed a new way to make more challenging humor detection datasets. We hypothesized that currently popular approaches, like discerning news or proverbs, can rely on recognizing domain-specific vocabularies instead of the semantic coherence that makes jokes funny. We illustrated this by constructing several models for humor detection on these new datasets; where we found that previous technologies indeed are not or barely able to distinguish jokes from similar non-jokes. For a more modern architecture like RobBERT, the performance is only slightly lower for the generated non-jokes. This shows that the generated negatives dataset is indeed more challenging, and that transformer models are a step in the right direction for humor detection given their context-sensitivity. These datasets and findings open the way for interesting new, more context-aware Dutch joke detection and generation algorithms.

Acknowledgements

References

1. Annamoradnejad, I.: Colbert: Using bert sentence embedding for humor detection. arXiv preprint arXiv:2004.12765 (2020)
2. van den Beukel, S., Aroyo, L.: Homonym detection for humor recognition in short text. In: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 286–291 (2018)
3. Branwen, G.: Gpt-3 creative fiction (2020)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
5. Caron, J.E.: From ethology to aesthetics: Evolution as a theoretical paradigm for research on laughter, humor, and other comic phenomena. Humor **15**(3), 245–281 (2002)

6. Cattle, A., Ma, X.: Recognizing humour using word associations and humour anchor extraction. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1849–1858 (2018)

7. Chen, P.Y., Soo, V.W.: Humor recognition using deep learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 113–117 (2018)

8. Deckers, L., Buttram, R.T.: Humor as a response to incongruities within or between schemata. Humor: International Journal of Humor Research (1990)

9. Delobelle, P., Cunha, M., Massip Cano, E., Peperkamp, J., Berendt, B.: Computational ad hominem detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 203–209. Association for Computational Linguistics, Florence, Italy (Jul 2019)

10. Delobelle, P., Winters, T., Berendt, B.: RobBERT: a Dutch RoBERTa-based language model. arXiv preprint arXiv:2001.06286 (2020)

11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)

12. Dodge, J., Gururangan, S., Card, D., Schwartz, R., Smith, N.A.: Show your work: Improved reporting of experimental results. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2185–2194. Association for Computational Linguistics, Hong Kong, China (Nov 2019)

13. Fürnkranz, J., Hüllermeier, E.: Pairwise preference learning and ranking. In: European conference on machine learning. pp. 145–156. Springer (2003)

14. Gibson, J.: A good sense of humor is a sign of psychological health. quartz (2016)

15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)

16. Hossain, N., Krumm, J., Gamon, M.: " president vows to cut¡ taxes¿ hair": Dataset and analysis of creative text editing for humorous headlines. arXiv preprint arXiv:1906.00274 (2019)

17. Kiddon, C., Brun, Y.: That's what she said: double entendre identification. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. pp. 89–94 (2011)

18. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014)

19. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs] (Jul 2019)

20. Mihalcea, R., Strapparava, C.: Making computers laugh: Investigations in automatic humor recognition. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 531–538. Association for Computational Linguistics (2005)

21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

22. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In: 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7). Cardiff, United Kingdom (Jul 2019)
23. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8) (2019)
24. Ritchie, G.: Developing the incongruity-resolution theory. Informatics Report Series (10 1999)
25. Suls, J.M.: A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis. The psychology of humor: Theoretical perspectives and empirical issues **1**, 81–100 (1972)
26. Taylor, J.M., Mazlack, L.J.: Computationally recognizing wordplay in jokes. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 26 (2004)
27. Tulkens, S., Emmery, C., Daelemans, W.: Evaluating unsupervised dutch word embeddings as a linguistic resource. In: Chair), N.C.C., Choukri, K., Declerck, T., Grobelnik, M., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)
29. Weller, O., Seppi, K.: Humor detection: A transformer gets the last laugh. arXiv preprint arXiv:1909.00252 (2019)
30. West, R., Horvitz, E.: Reverse-engineering satire, or "paper on computational humor accepted despite making serious advances". In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7265–7272 (2019)
31. Winters, T.: Generating philosophical statements using interpolated markov models and dynamic templates. In: 31st European Summer School in Logic, Language and Information Student Session Proceedings. pp. 181–189. Riga, Latvia, ESSLLI (Aug 2019)
32. Winters, T.: Generating dutch punning riddles about current affairs. 29th Meeting of Computational Linguistics in the Netherlands (CLIN 2019): Book of Abstracts (Jan 2019)
33. Winters, T.: Modelling mutually interactive fictional character conversational agents. In: Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) and the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019). vol. 2491. CEUR-WS (2019)
34. Winters, T., De Raedt, L.: Discovering textual structures: Generative grammar induction using template trees. Proceedings of the 11th International Conference on Computational Creativity pp. 177–180 (2020)
35. Winters, T., Nys, V., De Schreye, D.: Towards a general framework for humor generation from rated examples. Proceedings of the 10th International Conference on Computational Creativity pp. 274–281 (2019)
36. Yang, D., Lavie, A., Dyer, C., Hovy, E.: Humor recognition and humor anchor extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2367–2376 (2015)
37. Yeh, C.L., Loni, B., Hendriks, M., Reinhardt, H., Schuth, A.: Dpgmedia2019: A dutch news dataset for partisanship detection. arXiv preprint arXiv:1908.02322 (2019)

A   Dynamic Template parametrisation

The parameters used for the dynamic template algorithm to generate the non-jokes are a maximum word frequency of the 62% percentile, and minimum number of replacement of at least one replacement for every 25 characters, and three randomly sampled jokes for context words

B   Hyperparameter space

Table 2: The hyperparameter space used for the LSTM and CNN models with Dutch word embeddings.

Hyperparameter	LSTM model	CNN model
adam_epsilon	10^{-8}	10^{-8}
fp16	False	False
hidden_dimension	$i \in \{8, 16, 32, 64, 128\}$	—
learning_rate	$[10^{-3}, 10^{-1}]$	$[10^{-3}, 10^{-1}]$
pooling	—	max
convolutional_layers	—	2
kernel_size.	—	3
max_grad_norm	1.0	1.0
num_train_epochs	15	15
batch_size	64	64
seed	1	1
dropout	0.1	0.1

Table 3: The hyperparameter space used for finetuning RobBERT.

Hyperparameter	Value
adam_epsilon	10^{-8}
fp16	False
gradient_accumulation_steps	$i \in \{1, 2, 3, 4\}$
learning_rate	$[10^{-6}, 10^{-4}]$
max_grad_norm	1.0
max_steps	-1
num_train_epochs	3
per_device_eval_batch_size	8
per_device_train_batch_size	8
seed	1
warmup_steps	0
weight_decay	$[0, 0.1]$