Walter Burns at SemEval-2023 Task 5: NLP-CIMAT - Leveraging Model Ensembles for Clickbait Spoiling

Emilio Villa-Cueva[1] **Daniel Vallejo-Aldana**[1]
Fernando Sánchez-Vega[1] **Adrián Pastor López-Monroy**[1]
[1]Department of Computer Science, Mathematics Research Center (CIMAT), Guanajuato, Mexico
{emilio.villa,daniel.vallejo,fernando.sanchez,pastor.lopez}@cimat.mx

Abstract

This paper describes our participation in the *Clickbait challenge* at *SemEval 2023*. In this work, we address the Clickbait classification task using transformers models in an ensemble configuration. We tackle the Spoiler Generation task using a two-level ensemble strategy of models trained for extractive QA, and selecting the best K candidates for multi-part spoilers. In the test partitions, our approaches obtained a classification accuracy of 0.716 for classification and a *BLEU-4* score of 0.439 for spoiler generation.

1 Introduction

Clickbait is the name given to content whose primary purpose is to lure visitors to a web page. Often, this content is designed to appeal to users through sensationalism, aiming to affect their curiosity. Once the user takes the bait, very often contains a large text that has little to do with the promised content in the Clickbait, and there are only a few sentences that properly "answer" it. In their work, (D. Molina et al., 2021) describes that Clickbait texts rely on the cognitive phenomenon known as *Information Gap*, consisting of alerting the reader with some relevant information on the text but leaving a gap in the reader's knowledge hence triggering its curiosity and the need to fill this gap.

Although the purpose of clickbait is to appeal to the reader's curiosity and emotions, this can have side effects as well. As stated in (Zhou, 2022), clickbaits usually exaggerate headlines to attract the attention of the audience ignoring completely the fact that news should follow such as objectivity, truth and fairness. This can mislead the readers to have a wrong judgment of news events. The Clickbait Challenge at SemEval 2023 (Fröbe et al., 2023a) is a shared task whose objective is to develop systems that, given a Clickbait and the content of its linked page, can classify the required types of spoiler (task 1) and produce an adequate spoiler (task 2).

There are examples of the use of automatic methods for detecting Clickbaits (Potthast et al., 2016; Agrawal, 2016; Khater et al., 2018), in which they aim to develop a system that can identify Clickbait content and label it as such before it reaches the user.

However, there is less research on systems for automatically "answering" Clickbaits. In their work, Hagen et al. (2022) proposes "spoiling" the posts by treating the problem as either a passage-retrieval or a question-answering problem. They experimentally found there is a significant difference in favor of the question-answering approach. In the same work, the authors compile the "Webis Clickbait Spoiling Corpus 2022", a dataset that consists of 5000 manually spoiled posts, and define three possible categories of spoilers: *multi-part, passage,* and *phrase*.

In this work, we present our system for this shared task. For task 1, we fine-tune multiple RoBERTa models in a multi-task setting and multiple RoBERTa models in a binary task setting. We propose an attention neural ensemble architecture where all our pre-trained models are automatically weighted with attention mechanisms. For task 2, we approach the problem as extractive question-answering, we use an ensemble to improve the stability of the predictions given the relatively small dataset. Our ensemble approach considers two main aspects: (1) the use of different architectures, (2) averaging different instances of the architecture to improve stability, before extracting the final "answer". When evaluated in a validation partition, this two-level ensemble yields an improvement of around 11% when compared to a DeBERTa single model, which serves as a baseline.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 693–699
July 13-14, 2023 ©2023 Association for Computational Linguistics

2 Background

2.1 Spoiler type classification.

The problem of classifying clickbait spoilers has been addressed in different ways. (D. Molina et al., 2021) describes some features that a text should have to be potentially considered as clickbait such as the usage of questions, positive and negative superlatives, and modals.

In this work, we aim to automatically capture the features that lead to classifying a spoiler into three different types using Neural Networks models, especially state-of-the-art Transformers architectures.

Phrase spoilers are composed of a single word or phrase from the linked article, *passage* spoilers, include a set of –consecutive– sentences, while *multi-part* spoilers, on the other hand, are composed by several non-consecutive spoilers, making them the hardest to *spoil*.

For the spoiler type classification problem, we propose an ensemble architecture combining different types of fine-tuned RoBERTa (Liu et al., 2019) models trained to different tasks. As described in (Hagen et al., 2022), the RoBERTa-large configuration is a feasible approach for this problem, outperforming other state-of-the-art Transformer architectures. The advantage of using ensembles instead of a single model is described in (Guzman-Silverio et al., 2020), where a model ensemble vastly improved the performance of a model for aggressiveness detection in Twitter.

2.2 Clickbait spoiling as extractive QA.

The Clickbait spoiling task can be formulated as a question-answering problem. This is done by training a model to predict the "start" and "end" tokens of a given spoiler within a given web page linked by the Clickbait, which in a question-answering system is analogous to extracting the start-end token position of the answer in the context. However, since this task is not a traditional extractive QA problem, we must take into account two particularities: (1) for QA tasks, the context given to the BERT model is generally split into a couple of chunks due to the length limitations of the model[1]. In this case, the "context" is the concatenated title and body of the linked article. Since these are larger than common *context* sizes of extractive QA datasets (such as SQuAD), we end up with more chunks than the model was trained on, resulting in a more challenging setting. (2) The multi-part spoilers are found in different parts within the same context, this is challenging because extractive QA models are designed to extract a single interval. Because of this, a tailored strategy is necessary for these spoilers.

As noted by (Hagen et al., 2022), prior to fine-tuning the models for the Clickbait-spoiling task, we can take advantage of a language model that has already been fine-tuned in a question-answering dataset, such as SQuAD (Rajpurkar et al., 2016). In their work, they use models trained for SQuAD v1.1 due to the fact that we don't expect to find unanswerable Clickbaits. Nonetheless, we hypothesize that using SQuADv2.0 which contains unanswerable questions is far better for our particular application since for most examples the context will have to be split into chunks due to its length, thus, we want our model to discern chunks that contain useful information from those who don't.

Despite their performance, training large language models with a corpus of rather small size (5000) can lead to instability issues (Mosbach et al., 2021). This motivated us to utilize ensembles as a tool to assist in reducing these problems. By averaging the start-end predicted probabilities of several trained models, we effectively "smooth" the output distribution of the predictions, which in turn makes the predictions more robust.

Another factor is that the architecture and pre-training corpus of the model may output predictions biased to some degree, therefore, adding diversity by ensembling different architectures can reduce this. This can be done by computing scores for spoiler candidates using several models and then selecting the candidate with the highest score overall. A big advantage of this ensembling approach compared to averaging probabilities is that it is agnostic to the tokenizer. In our system, we utilize these two approaches to make the most of the base models.

3 System Overview

3.1 Spoiler classification

In this paper, we propose the attention neural ensemble, an architecture consisting of different RoBERTa-large (Liu et al., 2019) models trained with multi-class and binary settings. The different model configurations are then ensembled using an attention layer consisting of a Multi-Layer perceptron (Rosenblatt, 1958) to obtain the final

[1]For most language models, this limit is set to 512 tokens

predictions. This proposal aims to capture general features as well as specific features of each spoiler type with the multi-class and binary settings and then ensembling them to make a robust classification.

The attention neural ensemble model proposed in this work is illustrated in figure 1.

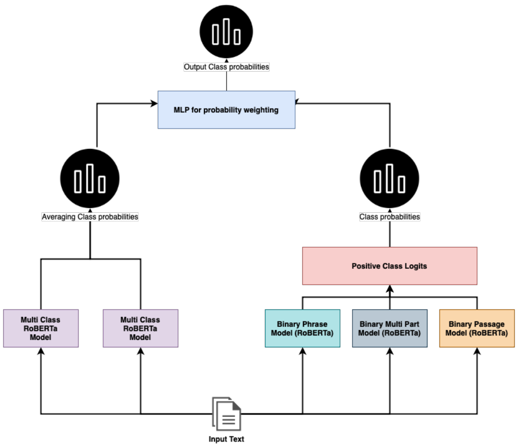

Figure 1: Attention Neural Ensemble for Clickbait spoiler Classification

To this aim, we first perform a pre-processing step over the input data (Hagen et al., 2022). From the data given, we only consider the Clickbait text and the Target Title, this last one consists of the title of the text from which the spoiler is generated. To standardize the input sequence length, we compute the length of each one of the input texts using a word tokenizer and truncate the size of the input sentences to be 95% quantile of the distribution of text lengths.

The first part of the classification pipeline proposed in this work consists of an ensemble of multiple multi-class models. To this aim, we use the RoBERTa-large (Liu et al., 2019) configuration. The reason is its performance in the multi-class classification task compared to smaller RoBERTa configurations and different Transformers architectures such as DeBERTa (He et al., 2020) and BERT (Devlin et al., 2018). The weights of the different model classification heads are randomly initialized. Each model is fine-tuned using the weighted Cross-Entropy as a loss function. The resulting logits are then normalized with the softmax activation function. To ensemble multiple classification models, we average these logits to obtain a final prediction.

Secondly, we combine different binary classification models. To achieve this, we train one classification model for each spoiler type (phrase, passage, and multipart). Then, we extract the logits of the positive class and combine them into a single vector that is normalized using a softmax activation function resulting in a multi-class probabilities vector. The binary model trained for multi-part spoiler classification is also used for spoiler generation.

Lastly, once the multi-class probability vectors are generated, we use a neural network to obtain a convex combination of the resulting vectors. The resulting vector is used for the final classification. This neural network corresponds to the attention layer illustrated in Figure 1.

3.2 Ensemble of extractive-QA models for spoiler generation

The main objective of the system is to generate spoilers that successfully satiate the curiosity induced by Clickbait, and we approach it as an extractive question-answering problem. During training time, each model in the ensemble is fine-tuned individually for the task of extractive question-answering.

Once a model has been through the two fine-tuning stages (SQuAD and then Clickbait-spoiling), we can extract candidate spoilers from the start-end probabilities generated by the model by assigning scores to the N most probable start-end pairs (spoiler candidates). Thus, for the inference stage, we implemented a two-level ensemble setup that takes advantage of several instances of architecture to enhance stability, and different architectures to provide variety in the predictions. As illustrated in Figure 2, when an input sample enters the system it passes through the two levels:

- The first level consists of several models of the same architecture, which have their start-end prediction logits averaged at a token level, the goal of this level is to stabilize the performance of these models. From the averaged logits in this step, we can proceed to extract a set of N-scored spoiler candidates from the context.

- The second level performs a union of all the sets of scored spoiler candidates provided by the M first-level ensembles. Since this level is agnostic to the tokenization procedure, it enables ensembling architectures with different tokenizers.

The pipeline returns a list of $N \times M$ scored spoiler candidates per example. As shown in Figure 2, we describe the configurations built using this system using the following notation: A first-level ensemble of –for example– two DeBERTa models (D) would be written as D^2. A second-level ensemble of two of the previous first-level ensembles would be written as $2D^2$, or if the other first-level ensemble is composed by a different architecture we would write it as $D^2 + R^2$ (R stands for RoBERTa), we use this notation in the results section, to describe the different tested configuration as minimally as possible.

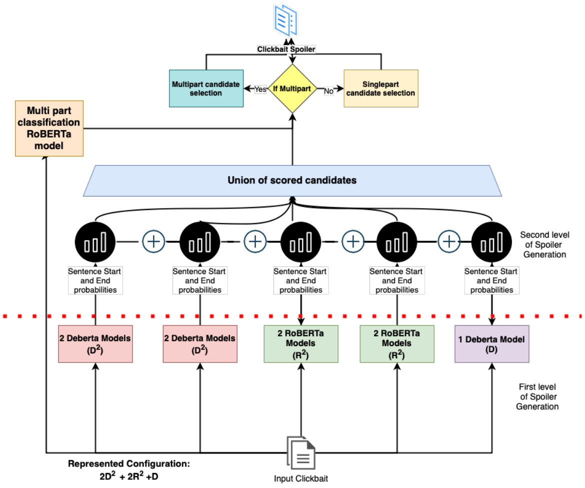

Figure 2: Diagram of the ensemble system for Spoiler Generation

Once the list of scored candidates has been generated, we must extract the best spoiler for each example. In this part, we take advantage of the predictions from task 1: if a spoiler has been predicted to be of type "passage" or "phrase", we simply take the candidate with the highest probability as the spoiler.

For "multi-part" spoilers, we extract K different spoilers from the candidates. This is done by taking the candidate with the highest score from the list and then setting its score (and the score of every candidate that intersects with it) to a very low value that disables it in the following iteration. This process is repeated K times, resulting in K non-intersecting spoilers retrieved from the spoiler candidates.

The intuition behind this is that a model trained for the extractive QA problem can output several candidates with different confidence scores, we presume that the different parts of a multi-part spoiler should be included in the list of scored spoiler candidates. In this work, we empirically select K as a fixed integer, a better procedure for selecting this quantity is left for future work.

4 Experimental Setup

The proposed systems were evaluated using the training and validation partitions provided in the competition: we found these partitions to be adequately split since class proportions were similar between the validation and training sets. We employ the same evaluation metrics used for the shared task. For classification, we evaluate the accuracy, and for spoiler generation, BLEU-4.

4.1 Training details for the classification models

We used pre-trained models available in the HuggingFace repository [2]. In the pre-processing stage, we consider a maximum length of quantile 95% of the distribution of the lengths over the dataset resulting in an input sequence length of 37 words.

For training, we set the batch size to 8, the optimizer used is AdamW (Loshchilov and Hutter, 2017) with a learning rate of $1e^{-5}$ and a weight decay factor of 0.1. The multi-class models were trained during 8 epochs whereas the binary models were trained during 5 epochs.

To train the attention neural ensemble, we froze all RoBERTa weights from the multiple models and trained only the final attention layer for 2 epochs with Adam (Kingma and Ba, 2014) optimizer with a learning rate of $1e^{-4}$.

4.2 Training details for the spoiler generation models

Based on the experimental results of (Hagen et al., 2022), we use RoBERTa and DeBERTa as base models for our approach. We used fine-tuned models available at the HuggingFace repository [3].

To increase the number of training examples for our model, we split the "multi-part" examples into individual "single-part" examples, this enlarged the training set by around 44%. We only perform this split in the training stage.

For training, we used an AdamW optimizer with a learning rate of $1e^{-5}$ and a batch size of 8, fine-tuning the models for 2 epochs. For both inference and training, we set the maximum input length to 512, with a stride –the number of tokens that overlap between chunks– of 128. Since most of the

[2] Web page `https://huggingface.co`

[3] deepset/deberta-v3-large-squad2, deepset/roberta-large-squad2, tli8hf/unqover-roberta-large-newsqa

examples were longer than 512 tokens, the majority of them were split into two or more chunks. The maximum number of candidates per example was set to N=40.

5 Results

5.1 Spoiler classification

To determine if different transformer architectures are able to classify Clickbait spoilers accurately, we conducted experiments evaluating different transformer configurations. We then examined if an ensemble of a base model increases its classification capability.

We tested different transformers architectures, RoBERTa (Liu et al., 2019), BERT (Devlin et al., 2018), and Deberta (He et al., 2020) for the spoiler classification task. For this purpose, we fine-tuned the models for the muli-class classification using the smallest configuration available. In Table 1, we show the performance of these models. We show the results of fine-tuning the models five times with different weights of initialization.

Model	Mean Accuracy	Min Accuracy	Max Accuracy
RoBERTa (Liu et al., 2019)	**70.0833**	**69.25**	**70.5**
BERT (Devlin et al., 2018)	67.2	67	67.5
Deberta (He et al., 2020)	69.49	68.65	70.5

Table 1: Results for multi-class classification over the validation dataset. It is worth noting that the RoBERTa architecture yields the best results over the validation dataset

Based on the previous experiments, we see that the RoBERTa (Liu et al., 2019) configuration shows better performance in multi-class spoiler classification. Hence, this configuration is used in this work for this task.

We evaluated different RoBERTa (Liu et al., 2019) configuration sizes. The results shown in Table 2 indicate that the large configuration of RoBERTa improves significantly the performance of the model.

Model	Mean Accuracy	Min Accuracy	Max Accuracy
RoBERTAa base	70.08	69.25	70.5
RoBERTa large	**73.44**	73.25	73.75

Table 2: Model comparison between different RoBERTa sizes

Lastly, we tested different proposals for multi-class classification using the RoBERTa large (Liu et al., 2019) configuration. The first one corresponds to an ensemble of fine-tuned models for multi-class classification. The second model corresponds to a combination of three different binary models trained to classify each one of the classes. Finally, we evaluate the attention neural ensemble used to obtain the results for task 1. The results shown in Table 3 correspond to the evaluation of the models tested over the validation dataset.

Model	Mean Accuracy	Min Accuracy	Max Accuracy
RoBERTa large	73.44	73.25	73.75
MultiClass Ensemble	74.16	74.05	74.25
Binary Models Combination	73.5	72.37	73.95
Attention Neural Ensemble	**74.37**	**74.25**	**74.53**

Table 3: Results of the proposed models over the validation dataset

As we see from Table 3, the attention neural ensemble showed a better performance compared to the rest of the tested models. This model was used to submit results for the challenge.

5.2 Spoiler generation

To assess how much we could improve performance for clickbait spoiling by scaling the two-level ensemble through different combinations of base models (which serve as a baseline) we experimented with several configurations (see Table 4). We ran these experiments were run with K=3, and observed that there is an important increase in performance by combining different architectures. Specifically, ensembling models in a two-level fashion achieve an important increase in BLEU-4 for passage and phrase spoilers, we can also notice that scaling the ensemble to a large number of models can improve performance for up to 9.5% (best ensemble compared to DeBERTa Baseline).

Thus, we selected $3D^2 + 2R^2 + R_N$ as our model[4] to submit. Although we observed a performance decrease in the BLEU-4 multi-part, we expected that adjusting the value of K could improve performance in this type of spoiler.

We empirically determine the best value for K, by evaluating the BLEU-4 performance of ensemble $3D^2 + 2R^2 + R_N$ using different values of K (See Figure 3). We observe that increasing K does improve the performance by an important degree (around 161%), this suggests that (1) using the multipart predictions of task 1 does improve performance in these spoilers, and (2) the framework

[4]Specifically, this configuration has three first-level ensembles of 2 DeBERTas trained on SQuAD 2, two first-level ensembles of 2 RoBERTas trained on SQuAD 2, and a RoBERTa trained on NewsQA, all of these first-level ensembles generate a list of candidates from which the spoiler is retrieved, at the second level of the ensemble

of extracting candidates by scores is adequate for the task. From our experiments, we conclude that, for the validation set, k=5 yields the best results overall.

System	BLEU-4	BLEU-4 passage
D (baseline)	$43.8_{0.79}$	$33.36_{1.85}$
R	$43.13_{0.81}$	$30.82_{1.19}$
D^2	43.85	33.87
R^2	44.24	32.73
$D^6 + R^4$	45.40	34.27
$D^2 + R^2$	45.33	34.32
$D^6 + R^4 + R_N$	45.85	34.40
$3D^2 + 2R^2$	46.20	35.04
$D^2 + R^2 + R_N$	45.13	32.57
$2D^2 + 2R^2 + R_N$	46.71	34.91
$3D^2 + 2R^2 + R_N$	**46.85**	**35.06**
	BLEU-4 phrase	BLEU-4 multipart
D (baseline)	$62.48_{0.18}$	$23.69_{1.26}$
R	$63.67_{1.27}$	$22.73_{1.35}$
D^2	62.16	23.42
R^2	64.02	23.79
$D^6 + R^4$	65.48	23.40
$D^2 + R^2$	64.77	**24.60**
$D^6 + R^4 + R_N$	66.57	23.10
$3D^2 + 2R^2$	66.74	23.21
$D^2 + R^2 + R_N$	66.61	23.10
$2D^2 + 2R^2 + R_N$	68.37	22.52
$3D^2 + 2R^2 + R_N$	**68.47**	22.77

Table 4: Results of different configurations on the validation set. D and R stand for DeBERTa and RoBERTa respectively (fine-tuned on SQuAD v2), and R_N stands for RoBERTa fine-tuned on NewsQA. The results of the single models (first two rows) that serve as a baseline, are the mean and standard deviation of 4 models.

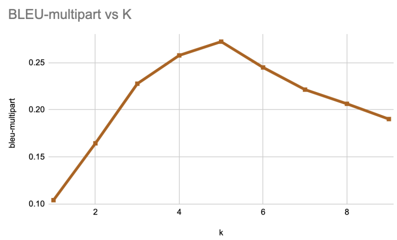

Figure 3: BLEU-4 (multi-part) in the validation set using different values of K

5.3 Results of the submitted system

Based on the results of the validation set, we trained models using full training, and validation sets used the same configuration as the best systems in the validation set. For spoiler generation, we used a $2D^3 + 2R^2 + R_N$ ensemble with $K = 5$. In Table 5 we can observe the final results of our submission in the test set and their performance compared to the mean of all participants. Our proposed system obtained a significantly greater performance in both subtasks when compared to the average performance of all participants. For task 1, we achieved the highest F1 score in detecting phrase spoilers. Furthermore, the two-level ensemble system used for task 2 ranked among the top 3 submissions and obtained the highest BERT score for extracting phrase spoilers. These results were submitted using the TIRA platform (Fröbe et al., 2023b) for evaluation.

	Result	Participant's average
Subtask 1 (balanced accuracy)	**0.716**	0.667
Subtask 2 (BLEU - 4)	**0.439**	0.311

Table 5: Comparison of the results obtained from our submissions and the average results obtained by the participants. We see that our proposals obtained a higher result than the average result showing the strengths of our proposals as spoilers classifiers and spoiler generators.

6 Ethical Issues

The creation of an automatic system capable of classifying the spoiler type and generating a spoiler from a query and a web page may help to reduce the amount of time a person spends trying to fill the information gap generated by the clickbait. However, entirely relying on an automatic system may generate another information gap as well by retrieving an incomplete answer, and more importantly, if the system fails to retrieve the correct information, the result can mislead the reader to rely on fake information. To avoid these issues, a more robust automatic spoiler generation system must be designed to retrieve a correct and complete answer independent of the spoiler type and particular text features.

7 Conclusion

This paper described our participation in the Clickbait Challenge at SemEval 2023. Our results demonstrate that ensembling techniques yield a performance increase in both classification and spoiler-generation tasks compared to baseline models.

For classification, we empirically demonstrate that the combination of different classification models improves accuracy by an important degree compared to just fine-tuning a single model.

For the clickbait-spoiling task, we note that the two-level ensemble configuration produces adequate results by addressing the issues of stability and bias of the base models. Furthermore, the approach taken on multipart spoilers of taking a set of potential candidates produces better spoilers compared to only taking the best candidate. A dynamic selection of the K parameter depending on the context is left for future work.

In conclusion, our approach based on ensembles for both tasks successfully robustly tackles them, resulting in adequate results compared to baseline models.

8 Acknowledgments

The authors thank CONACYT, INAOE, and CIMAT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies (Laboratorio de Supercómputo: Plataforma de Aprendizaje Profundo) with the project *"Identification of Aggressive and Offensive text through specialized BERT's ensembles"* and CIMAT Bajio Supercomputing Laboratory (#300832).

References

Amol Agrawal. 2016. Clickbait detection using deep learning. In *2016 2nd international conference on next generation computing technologies (NGCT)*, pages 268–272. IEEE.

Maria D. Molina, S Shyam Sundar, Md Main Uddin Rony, Naeemul Hassan, Thai Le, and Dongwon Lee. 2021. Does clickbait actually attract more clicks? three clickbait studies you must read. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Maik Fröbe, Tim Gollub, Benno Stein, Matthias Hagen, and Martin Potthast. 2023a. SemEval-2023 Task 5: Clickbait Spoiling. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*.

Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023b. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.

Mario Guzman-Silverio, Ángel Balderas-Paredes, and Adrián Pastor López-Monroy. 2020. Transformers and data augmentation for aggressiveness detection in mexican spanish. In *IberLEF@ SEPLN*, pages 293–302.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait Spoiling via Question Answering and Passage Retrieval. In *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, pages 7025–7036. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Suhaib R Khater, Oraib H Al-sahlee, Daoud M Daoud, and M Samir Abou El-Seoud. 2018. Clickbait detection. In *Proceedings of the 7th international conference on software and information engineering*, pages 111–115.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016. Clickbait detection. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 810–817. Springer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Zhi Zhou. 2022. The analysis of click bait in we media era. In *2021 International Conference on Social Development and Media Communication (SDMC 2021)*, pages 1544–1547. Atlantis Press.