

# Adversarial Question Answering in Spanish with Transformer Models

Alejandro Vaca-Serrano<sup>1</sup>

<sup>1</sup>*Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th floor UAM Cantoblanco, 28049 Madrid, Spain*

## Abstract

In this work, a system for adversarial Question Answering (QA) in Spanish is presented. Models BETO, MarIA-base, MarIA-large and RigoBERTa are tried, although finally only last 3 are used. They are first trained over a big adversarial QA corpus, AllQA. AllQA is composed of SQUAD-ES v2, a translated version of NewsQA presented in this work, and QUALES. These general QA models are then retrained over QUALES dataset. Finally, their predictions are aggregated via meta-ensembling techniques, to produce more reliable answers to the presented questions. Results in terms of F1-score are presented on the validation set of AllQA and QUALES, and complete official results on the test set of QUALES in terms of F1-score and exact match are also presented.

## Keywords

Transformers, Question Answering, Spanish Models, Meta-Ensemble

## 1. Introduction

In this work, different Transformer-based solutions are explored for answering questions in an extractive way, with some questions not having any answer (adversarial question answering). This format is similar to that of SQuAD v2 [1], which is an augmented version of SQuAD v1 [2], the most known and used extractive question-answering dataset up to date. First of all, previous related work is reviewed in section 2, then, tasks are described in section 3. Models are presented in sections 4 and 5, together with their evaluation results in section 6. Finally, in section 7, conclusions and future work are presented.

## 2. Related Work

### 2.1. Extractive Question Answering Datasets in Spanish

As stated in the previous section, SQuAD [2] is the biggest openly-available extractive question answering dataset. Its v1 has 100,000+ questions-answer pairs, posed by crowdworkers on a set of Wikipedia articles. On the other hand, for v2, over 50,000 unanswerable questions are added to the already existing ones. As will be clear when exploring the resources available in Spanish,

---


*IberLEF 2022, September 2022, A Coruña, Spain.*

✉ [alejandro\\_vaca0@hotmail.com](mailto:alejandro_vaca0@hotmail.com) (A. Vaca-Serrano)

🌐 <https://www.linkedin.com/in/alejandro-vaca-serrano/> (A. Vaca-Serrano)



© 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)

extractive question answering datasets in Spanish have significantly less instances, therefore reducing the capacity of Spanish question answering models compared to the English ones.

In the recent years there has been an increased effort to create question answering datasets in other languages than English, the language in which most extractive question answering (QA) datasets are developed. One such effort is MLQA [3], a dataset developed due to the scarcity of QA datasets in languages other than English, and intended to evaluate cross-lingual models. MLQA has over 12K instances in English and 5k in each of the other languages (Spanish, Arabic, German, Hindi, Vietnamese and Simplified Chinese).

A similar approach, although only for Spanish and using Automatic Translation, is the dataset SQuAD-ES [4]. This dataset is an automatic translation of SQuAD (both versions 1 and 2) to Spanish. Although it is not expected that its quality is as high as a dataset originally developed in Spanish, it takes advantage of the fact that modern automatic translators are of good quality, and the English version of SQuAD has many instances.

Recently, the first high-quality, high-volume QA dataset was originally developed in Spanish, SQAC (Spanish Question Answering Corpus) [5]. Although it does not provide adversarial examples, it is relevant in the sense that no dataset of a similar size originally in Spanish was available prior to its release. However, and even though this is a great advance for the Spanish NLP community, its size is still very clearly behind those in other languages. SQAC contains 6,247 contexts and 18,817 questions with their answers.

Taking the French language as an example, which has less speakers than the Spanish language, the FQuAD [6] dataset is composed of 60,000+ questions and answer samples. Moreover, it has an adversarial version, FQuAD 2.0 [7]. Both of them are originally developed in French. This comparison highlights the existing gap between Spanish and other languages in terms of resources for building strong QA systems.

## 2.2. Transformer Models in Spanish

The first language model released in Spanish was BETO [8], a Spanish BERT [9]. Then, in the context of the MarIA project [5], Spanish RoBERTa [10] and GPT-2 [11] models were released, both base and large. As we are only interested in encoder-based models for this work, due to the nature of QUALES task, RoBERTa-base model will be referred to as MarIA-base, while RoBERTa-large will be called MarIA-large. Additionally, BERTIN model was released this year [12]. It is also a version of RoBERTa in Spanish, trained with less resources than [5] but with novel techniques. Finally, RigoBERTa [13] was released this year. It is a Spanish DeBERTa model [14] which performs generally better than the rest of the language models in Spanish, as shown in [13]. It is the best performing model in 10 out of 13 tasks including classification, NER and QA tasks.

In [13] authors explain why it is not possible to use BERTIN [12] for QA tasks, due to a bug in its tokenizer. For that reason, that model is automatically discarded for this task, as it is not possible to use it with the same preprocessing procedure as the rest of the models.

Finally, models BETO [8], MarIA-base, MarIA-large [5] and RigoBERTa [13] are used for this task.

Subset	Percentage of Unanswerable Answers
Train	0.173
Development	0.169
Test	0.134

**Table 1**  
Percentage of unanswerable answers per subset.

### 3. Task Description

The task is the standard adversarial QA task. This means that systems are given a question and a context, and they need to detect the context spans that answer that question. Unanswerable questions have to be answered with an empty string. In this case, systems cannot be of generative nature, as the submitted answers must be in the provided contexts. This restricts the type of models that can be used for this task.

In the case of QUALES, the percentage of unanswerable questions is significantly lower than that of SQuAD, for example. Table 1 shows the percentage of unique unanswerable answers in each of the known subsets. Another difference with SQuAD or other datasets of this type is that several questions had multiple answers. This means that, given the same question and context, systems learn over different answers, with no change in the input.

For assessing the distribution similarity between the known subsets of the dataset and the unknown test subset, and given that Codalab is used for this challenge, a full null submission was sent. A full null submission has an empty string for all answers. In this way, it is possible to compute the percentage of answers in the test set that are unanswerable. This data is also reflected in table 1, in which subtles differences between subsets can be appreciated. This means that systems are trained over a distribution of unanswerable answers different to the one over which they are tested.

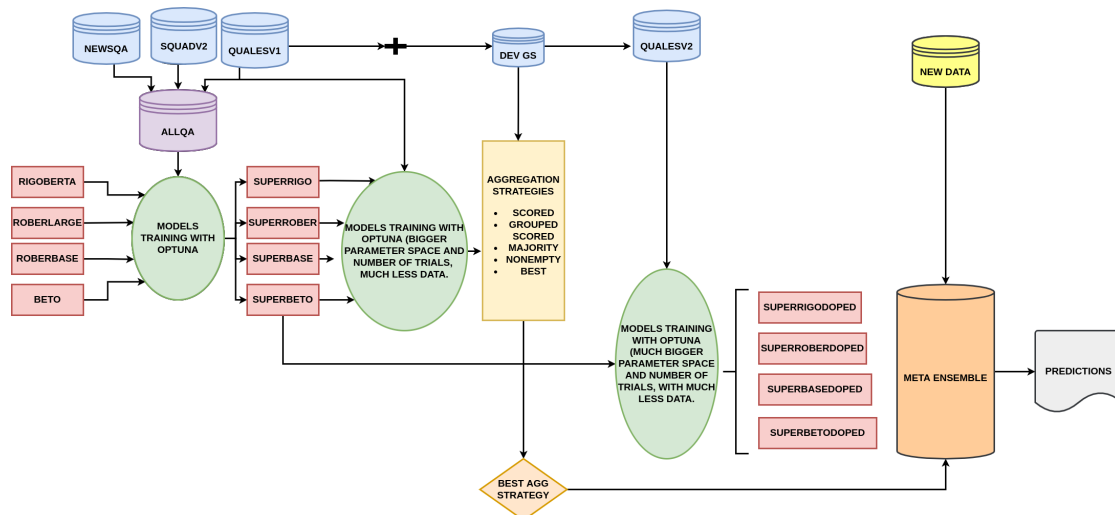
### 4. System Description

As there was not much training data for this task, it was decided to train the models in two phases. First, models are trained over a big QA dataset composed of multiple ones. This dataset will be called AllQA along the paper, and its creation will be explained in the next section. Then, all models are trained over this dataset, in order to have models that are able to answer general questions, although they are not adapted to the challenge data yet.

After that, these models are retrained with the competition data. Only good performing models are kept at this point. With the good performing models obtained from this last step, predictions are carried out, and those predictions are mixed as it is explained in further sections.

This is done with the first version of QUALES dataset (QUALES v1), that is, the one only containing the training set. When the development set gold standard was released, it was used to try and select the best aggregation strategies (more information below) for all models' predictions.

Finally, all QUALES data was put together, randomly sampling a 15% of it as the development



**Figure 1:** Graphic description of the whole system. Models are first trained on AllQA dataset. Then, they are retrained over QUALES v1. The development set gold standard was used to optimize the aggregation strategies, which mix the models’ predictions to produce a final prediction for each sample. After that, general QA models are retrained with QUALES v2, which contains all training and development set, with 85% for training and the rest for validation. With these predictions and the aggregation strategy selected from previous step, final predictions are produced when new data is presented.

set. Additionally, samples that would be in the validation split but were already included in the training set of AllQA are taken out of the validation set and added to the training set.

This second version of QUALES will be called QUALES v2 along the paper. General QA models are retrained with QUALES v2, then using the best aggregation strategy to mix their predictions, thus producing the final predictions for the competition. The whole system is depicted in figure 1.

#### 4.1. AllQA dataset creation

AllQA dataset is created by putting together a version of SQUAD v2 [1] in Spanish [4], a translated version of NewsQA [15], and the competition data, although this last one represents a very small percentage of the final AllQA dataset. Other approaches were tried, such as using SQAC [5], a translated version of BioAsq [16] and a translated version of COVID-QA [17], adding adversarial examples via random sampling, but they were discarded as worked worse than the presented approach. Also, as these three datasets were not originally developed as adversarial QA datasets, noise could be included by artificially creating adversarial examples via random sampling. Therefore, only original adversarial QA datasets were included for this step.

##### 4.1.1. Translation and Answer Matching for the NewsQA dataset

For translating the NewsQA dataset, Helsinki translation models were used to translate from English to Spanish [18]. Three fields are translated: question, context and answer text. As this

Hyperparameter	Values
Learning Rate	(5e-6, 5e-5, log)
Num Train Epochs	{2, 3, 5, 7}
Train Batch Size	{16, 32, 48, 64, 128}
Warmup Steps Ratio	(0.01, 0.10, log)
Weight Decay	(1e-3, 0.3, log)
Adam Epsilon	(1e-10, 1e-6, log)

**Table 2**  
Hyperparameter space for AllQA large models.

is a Machine Learning Approach based in Transformers [19] translated answers are not exactly the same as the part of the translated contexts that contain those answers.

Answers texts must be in the context, so a further processing step is needed. For each sample in the dataset, the following is carried out. Answer text and context are splitted with spaces. Then, and only when the answer is not the empty string, windows of tokens from the context are created, from  $window\_size = 1$  to  $window\_size = 200$ . That is, answers are assumed to be between 1 and 200 words long.

Each of those context windows are compared against the answer words list, computing the intersection of words between them. The higher the intersection between a context window and the answer words list, the more points for that context window. Additionally, the absolute difference between both lists' lengths is subtracted from those points, as we trust more context windows that apart from having many words in common with the answer words list, also have a similar length. Finally, extra points are added in case the context window and the answer words list start and end with the same words (2 points each).

After NewsQA is translated and processed, the following is done. As NewsQA [15] and SQUAD-ES v2 [4] have a proportion of adversarial examples that differs very much with the proportion of adversarial examples of QUALES [20], a random sampling of the adversarial examples was done in each case to match the proportion of adversarial examples for QUALES in the training set. In the end, AllQA is created split by split, that is, the training set is composed of the training set of each dataset. The validation split is created by subsampling a 10% of the training set, and the validation set of each dataset, including QUALES, is used as the test set for AllQA.

## 5. Models Training

### 5.1. General QA Models Training

Models are then trained over AllQA dataset described above. All models in the paper are trained with the use of Huggingface Transformers [21]. For that, Optuna [22] is used to optimize the hyperparameters for those models. The hyperparameter space for the base and large models are presented in tables 3 and 2 respectively. 20 trials in total (first 10 are random) are carried out with each model, saving the checkpoint in each case that performs best on the development set. The metric used for selecting the best checkpoint in each case is the eval loss.

Hyperparameter	Values
Learning Rate	(5e-6, 8e-5, log)
Num Train Epochs	{2, 3, 5, 7}
Train Batch Size	{16, 32, 48, 64, 128}
Warmup Steps Ratio	(0.01, 0.10, log)
Weight Decay	(1e-3, 0.3, log)
Adam Epsilon	(1e-10, 1e-6, log)

**Table 3**  
Hyperparameter space for AllQA base models.

Hyperparameter	Values
Learning Rate	(5e-6, 5e-5, log)
Num Train Epochs	{3, 5, 7, 10, 15, 20, 25, 30}
Train Batch Size	{16, 32, 48, 64, 128}
Warmup Steps Ratio	(0.01, 0.10, log)
Weight Decay	(1e-3, 0.3, log)
Adam Epsilon	(1e-10, 1e-6, log)

**Table 4**  
Hyperparameter space for QUALES v1 large models.

## 5.2. QUALES V1 Models Training

With models from last step, further training is carried out. The reason behind this is that models are expected to learn the general aspects of the task from the last step, but they are not specific to this corpus, which, as explained previously, has several differences with any other QA dataset, such as having different answers for the same question-context pair. For this reason, this second step is used to adjust models to the domain and specifics of the QUALES task.

Not all models are used for this second step. Given the bad results of BETO on AllQA (see 6), it was discarded, so only MarIA-large, MarIA-base and RigoBERTa are used (BETO results are also reported). These are called Superrober, Superbase and Superrigo in figure 1.

Hyperparameters used for models in this second step are in tables 5 and 4 for base and large models, respectively. In this case 120 trials were carried out in each case using Optuna [22], with the first 25 trials being random, so that enough exploration in the parameter space was carried out before optimizing.

## 5.3. Aggregation Strategies

With models trained on QUALES v1, different aggregation strategies are used. Aggregation Strategies are different ways of mixing the answers of the models to create a final answer. It can be seen as a kind of meta-ensemble. The different aggregation strategies considered are described below.

Hyperparameter	Values
Learning Rate	(5e-6, 8e-5, log)
Num Train Epochs	{3, 5, 7, 10, 15, 20, 25, 30}
Train Batch Size	{16, 32, 48, 64, 128}
Warmup Steps Ratio	(0.01, 0.10, log)
Weight Decay	(1e-3, 0.3, log)
Adam Epsilon	(1e-10, 1e-6, log)

**Table 5**  
Hyperparameter space for QUALES v1 base models.

### 5.3.1. Grouped Scored

Answer texts are grouped, aggregating them and counting the number of models that predicted each. So, for example, if for a given sample 2 models predicted answer "21" and other model predicted the null answer, a new answers dictionary is created with this form: {"21": 2, "": 1}.

This way, we weight predictions by the number of models that have predicted them. We use these weights for the next step. Validation scores are used so that each model is assigned a weight of 0.8 to 1.0, by scaling the scores for the models on validation. Answers scores from the last step are multiplied by the scaled weight assigned to the model predicting that answer.

Those scores assigned to each answer are later multiplied by the maximum between 1 and the length of the prediction in words (splitted by spaces). This gives priority to longer questions, and was designed to avoid too many null answers.

Moreover, these scores are multiplied by the logit score assigned by each model to those answers, so that the models' confidence in their answers is also taken into account.

Also, it was detected that some predictions are repeated a lot by the models. For that reason, the scores for new predictions (not predicted for other samples) are multiplied by 1.0, while the scores for already predicted answers are multiplied by 0.7.

### 5.3.2. Scored

In this case only validation split scores and logit scores are taken into account. The validation score for each model is multiplied by the logit assigned by it. Then, using these values, the prediction with the highest value is selected. This uses only our confidence in each model (represented by their validation score) and the confidence each model has on their prediction (represented by the logit score assigned to their answers).

### 5.3.3. Majority

In this case predictions are grouped, as in the Grouped Score strategy. If there is any answer which has been predicted by more than one model, it is selected as the final answer. Otherwise, the prediction of the model with the highest validation score is used.

**Table 6**  
F1-Score Results for AllQA.

Model	F1
RigoBERTa	<b>55.03</b>
MarIA-large	53.02
MarIA-base	54.03
BETO	44.51

#### 5.3.4. Non-empty

In this case only non-empty answers are considered. In case there is no non-empty answer for a given sample, the empty string is the final answer. Otherwise, answers that are not empty are weighted by the validation scores the models producing them obtained, choosing the answer from the model with the highest validation score.

#### 5.3.5. Best

This is the simplest strategy, as it only involves choosing the answer from the model that obtained the highest validation score, therefore no real mixing is carried out in this case.

In the end, the Grouped Scored strategy is used, as it provides better results than the rest of the methods on the Development Set Gold Standard.

### 5.4. QUALES v2 Models Training

When both training and development sets were available for QUALES, QUALES v2 was created, which concatenates both splits and resplits them, randomly selecting 15% for validation purposes. With this augmented dataset, models were trained again, as with QUALES v1. In fact, the same hyperparameter spaces presented in tables 5 and 4 are used for base and large models, respectively. The only difference was that in this case a total of 200 trials were carried out with Optuna [22], with 40 random initial trials.

## 6. Results

### 6.1. Results for AllQA

For AllQA, results are measured on the validation set. Datasets library [23] is used for evaluating the results, which contains SQUAD v2 metrics [1] such as exact-match and f1-score. F1-score is reported, as exact match is more biased, in the sense that a point after the answer, for example, does not change the nature of the answer; the doubt is solved if what comes before the point is correct. However, it would not count for exact match. Table 6 contains the results in terms of F1-score for all models trained on this task.



**Table 7**  
F1-Score Results for QUALES v1.

Model	F1
RigoBERTa	<b>65.94*</b>
MarIA-large	65.81
MarIA-base	61.18
BETO	51.24

**Table 8**  
F1-Score Results for the Validation Set of QUALES v2.

Model	F1	F1 with optimized prediction parameters
RigoBERTa	<b>72.85*</b>	<b>84.79*</b>
MarIA-large	69.25	84.06
MarIA-base	69.17	83.19
BETO	59.04	-

## 6.2. Results for QUALES V1

Models trained on QUALES v1 are evaluated on the subset randomly taken out for validation during the training phase. These results are reflected in table 7.

## 6.3. Results for QUALES V2

Finally, models trained on QUALES v2 are evaluated on the validation subset of that dataset. The results on this dataset are presented in table 8. In this case results are better, as more training data was used.

## 6.4. Official Test Results

MarIA-large, MarIA-base [5] and RigoBERTa [13] models trained on QUALES v2 are used for getting the final predictions (SuperRoberDoped, SuperBaseDoped and SuperRigoDoped in figure 1). As shown in figure 1, their predictions are mixed using the Grouped Score Aggregation Strategy presented previously.

Before getting the final predictions, additional optimizations for the models are carried out. The first one consists on setting an optimum maximum score for the null prediction score. A brute-force search is carried out with each model, by setting the maximum null score value to a value in the range of -15 to 15, in 0.5 step size. F1-score is computed in each case and the value maximizing it in the development set is the one chosen.

The same is done with maximum possible answer length and maximum possible start and end logits considered. In fact, these three parameters are optimized at the same time, so that the best combination of them is chosen for each model. As can be seen in the third column of table 8, results on the validation set are significantly improved by optimizing these parameters.

**Table 9**

F1-Score Results for the Official Test Set of QUALES challenge.

Exact Match	F1
39.92	58.77

Table 9 presents the official Exact-Match and F1-Score results for the system described in this paper. It obtains the fourth best results in terms of f1-score, although, as seen in table 9, results are worse than expected when looking at tables 8 and 7.

## 7. Conclusions and Future Work

In this work an adversarial QA system was presented, which uses three models in Spanish for the QA task: MarIA-base, MarIA-large [5] and RigoBERTa [13]. These were trained on AllQA dataset, created in this work by automatically translating NewsQA [15], among other tasks. After that, these general QA models are trained on QUALES, first on v1 and then on v2. Aggregation strategies are analyzed to mix the models' answers. The best aggregation strategy is Grouped Scored, described above. Prediction parameters such as maximum possible null score, max answer length and maximum number of considered start and end logits are optimized based on f1-score for each model. This significantly improves the performance of these models as seen in table 8.

As for future work, instead of mixing the models' predictions, an Ensemble model could be built, which uses the three selected Encoder-based models as the base models, and then internally mixes their last layers to produce more reliable predictions. Another possibility to improve the performance would be to remove duplicated question-context pairs. Additionally, there is a clear gap between the performance obtained in the validation set and the performance over the official test set; therefore this difference should be investigated to analyze the reasons for such a gap.

It is clear from the results tables that the best performing model, in general, for the adversarial QA task in Spanish, is RigoBERTa [13]. This coincides with the results presented in [13], where authors find that RigoBERTa is the best performing model on all QA tasks. This is reasonable since RigoBERTa is based on the DeBERTa architecture [14], which accounts better for the positional information in texts, crucial for this task.

## References

- [1] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for squad, CoRR abs/1806.03822 (2018). URL: <http://arxiv.org/abs/1806.03822>. arXiv:1806.03822.
- [2] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, CoRR abs/1606.05250 (2016). URL: <http://arxiv.org/abs/1606.05250>. arXiv:1606.05250.

- [3] P. S. H. Lewis, B. Oguz, R. Rinott, S. Riedel, H. Schwenk, MLQA: evaluating cross-lingual extractive question answering, *CoRR abs/1910.07475* (2019). URL: <http://arxiv.org/abs/1910.07475>. arXiv:1910.07475.
- [4] C. P. Carrino, M. R. Costa-jussa, J. A. R. Fonollosa, Automatic Spanish Translation of the SQuAD Dataset for Multilingual Question Answering, *arXiv e-prints* (2019) arXiv:1912.05200v1. arXiv:1912.05200v2.
- [5] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodríguez-Penagos, A. Gonzalez-Agirre, M. Villegas, Maria: Spanish language models, *Procesamiento del Lenguaje Natural* 68 (2022) 39–60. URL: <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6405>.
- [6] M. d’Hoffschmidt, M. Vidal, W. Belblidia, T. Brendlé, Fquad: French question answering dataset, *CoRR abs/2002.06071* (2020). URL: <https://arxiv.org/abs/2002.06071>. arXiv:2002.06071.
- [7] Q. Heinrich, G. Viaud, W. Belblidia, Fquad2.0: French question answering and knowing that you know nothing, 2021.
- [8] J. e. a. Cañete, Spanish pre-trained bert model and evaluation data, 2020. URL: <https://users.dcc.uchile.cl/~jperez/papers/pml4dc2020.pdf>.
- [9] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, *CoRR abs/1810.04805* (2018). URL: <http://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [10] Y. e. a. Liu, Roberta: A robustly optimized bert pretraining approach, 2019. URL: <https://arxiv.org/pdf/1907.11692.pdf>.
- [11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners (2018). URL: <https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf>.
- [12] J. D. la Rosa y Eduardo G. Ponferrada y Manu Romero y Paulo Villegas y Pablo González de Prado Salas y María Grandury, Bertin: Efficient pre-training of a spanish language model using perplexity sampling, *Procesamiento del Lenguaje Natural* 68 (2022) 13–23. URL: <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6403>.
- [13] A. Vaca Serrano, G. G. Subies, H. M. Zamorano, N. A. Garcia, D. Samy, D. B. Sanchez, A. M. Sandoval, M. G. Nieto, A. B. Jimenez, Rigoberta: A state-of-the-art language model for spanish, 2022. URL: <https://arxiv.org/abs/2205.10233>. doi:10.48550/ARXIV.2205.10233.
- [14] P. e. a. He, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: <https://arxiv.org/pdf/2006.03654.pdf>.
- [15] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, K. Suleman, Newsqa: A machine comprehension dataset, *CoRR abs/1611.09830* (2016). URL: <http://arxiv.org/abs/1611.09830>. arXiv:1611.09830.
- [16] A. Nentidis, A. Krithara, K. Bougiatiotis, M. Krallinger, C. R. Penagos, M. Villegas, G. Paliouras, Overview of bioasq 2020: The eighth bioasq challenge on large-scale biomedical semantic indexing and question answering, *CoRR abs/2106.14618* (2021). URL: <https://arxiv.org/abs/2106.14618>. arXiv:2106.14618.
- [17] T. Möller, A. Reina, R. Jayakumar, M. Pietsch, Covid-qa: A question answering dataset for covid-19, in: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, 2020.

- [18] J. Tiedemann, S. Thottingal, OPUS-MT – Building open translation services for the World, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 2020.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: <http://arxiv.org/abs/1706.03762>. arXiv:1706.03762.
- [20] A. Rosá, L. Chiruzzo, L. Bouza, A. Dragonetti, S. Castro, M. Etcheverry, S. Góngora, S. Goycochea, J. Machado, G. Moncecchi, J. J. Prada, D. Wonsever, Overview of QuALES at IberLEF 2022: Question Answering Learning from Examples in Spanish, *Procesamiento del Lenguaje Natural* 69 (2022).
- [21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface’s transformers: State-of-the-art natural language processing, CoRR abs/1910.03771 (2019). URL: <http://arxiv.org/abs/1910.03771>. arXiv:1910.03771.
- [22] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, CoRR abs/1907.10902 (2019). URL: <http://arxiv.org/abs/1907.10902>. arXiv:1907.10902.
- [23] Q. Lhoest, A. Villanova del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, J. Davison, M. Šaško, G. Chhablani, B. Malik, S. Brandeis, T. Le Scao, V. Sanh, C. Xu, N. Patry, A. McMillan-Major, P. Schmid, S. Gugger, C. Delangue, T. Matussière, L. Debut, S. Bekman, P. Cistac, T. Goehringer, V. Mustar, F. Lagunas, A. Rush, T. Wolf, Datasets: A community library for natural language processing, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 175–184. URL: <https://aclanthology.org/2021.emnlp-demo.21>.