Automatic Classification of the Economic Activity of a Company Using ML and DL Techniques

Roi Santos-Ríos

Universidade da Coruña, Centro de Investigación CITIC - Grupo LYS, Facultad de Informática, Campus de Elviña, 15071 - A Coruña (España)

Abstract

We present in his work our contribution to the CIDMEFEO project, developed in collaboration with the *Spanish National Statistics Institute* (INE). Our work focuses on the development of a text classification prototype for the identification and labeling of the different economical activities performed by Spanish companies. Such classification is made according to the so-called CNAE standard, which defines 629 hierarchically-ordered economical activities, taking as input the descriptions given by the companies. The great variability of the length and quality of these descriptions, together with the unbalanced nature of the datasets available for the task, make this task very difficult.

Three types of approaches are considered: a FastText-based system (to be used as baseline); a transformer-based approach (i.e. using BERT-style models); and finally, a LLM-based solution.

Keywords

Natural Language Processing, Text Classification, Machine Learning, Data Generation

1. Introduction

The work presented in his article is part of the CIDMEFEO project, developed in collaboration with the Spanish National Statistics Institute (INE, for *Instituto Nacional de Estadística*).[1] The main purpose of our research line within the project is the development of a NLP-based automatic encoder for the identification and correct labeling of the different economical activities that companies can perform. This classification is made according to the so-called CNAE standard (from *Clasificación Nacional de Actividades Económicas*), that defines 629 hierarchically-ordered economical activities, so that similar ones have similar codes.

This process takes as input a brief text provided by the company describing its activity. Our goal is to create a system that is capable of automatically assigning the corresponding CNAE code by interpreting such briefing. To do this, we have had to extensively review the state-of-the-art regarding NLP, specifically focusing on Text Classification techniques, as well as getting acquainted with the data provided by INE.

2. Background

Text classification, a core task in NLP, has advanced significantly with both traditional and modern methods. Earlier approaches such as Naive Bayes [1], SVM [2], and decision trees [3] relied on handcrafted features and bag-of-words or TF-IDF models. These techniques performed reasonably well, but often struggled with understanding the deeper semantics of language. FastText [4], developed by Facebook AI, improved on these by using subword information and word embeddings, offering faster and more efficient classification, especially for large datasets. However, while it enhanced generalization, it still fell short in handling complex contextual meanings.

Doctoral Symposium on Natural Language Processing, 25 September 2025, Zaragoza, Spain.

roi.santos.rios@udc.es (R. Santos-Ríos)
https://dunque.github.io/ (R. Santos-Ríos)
0009-0009-0541-1232 (R. Santos-Ríos)

[1] https://www.ine.es

The introduction of modern neural networks, particularly recurrent neural networks (RNN) and convolutional neural networks (CNN), further improved text classification by capturing sequential and local patterns in text. However, transformer-based models such as BERT [5], RoBERTa [6], and GPT [7] revolutionized the field by learning contextualized word representations and handling long-range dependencies. Pre-trained on massive datasets, these models have become the state-of-the-art, delivering superior performance across various classification tasks, from sentiment analysis to topic categorization, with minimal fine-tuning. Their ability to generalize with limited labeled data has established transformers as the leading architecture in NLP.

3. Materials and Methods

Three different approaches have been considered: a baseline classifier using FastText; a more advanced Deep Learning (DL) based approach using transformers (BERT-type models); and finally, a system using Large Language Models (LLM), both small local models and API-based ones.

However, before continuing, we first need to have a deep look at our working dataset, analyzing it and applying the necessary preprocessing techniques in order to prepare it for the models to be used.

3.1. Dataset

As explained above, the CNAE (from *Clasificación Nacional de Actividades Económicas*) is the Spanish standard classification system for economic activities. Most of its structure is inherited from the European Community standard, the so-called *Statistical Classification of Economic Activities* (NACE, from its French initials). The current work version is CNAE-2009,[2] that is structured in four hierarchical levels, as shown in Figure 1:

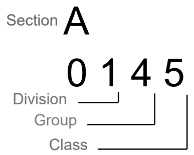

Figure 1: Example of the hierarchy of a CNAE code.

1. **Section**: represented by a letter (21 categories).
2. **Division**: represented by 2 digits (88 categories).
3. **Group**: represented by 3 digits, and where the first 2 digits indicate the corresponding division (272 categories).
4. **Class**: represented by 4 digits, and where the first 3 digits indicate the corresponding group (629 categories).

The regulations state that every company must be associated with a single category on each level. This means that, if the company develops more than one economic activity, the most relevant one must be chosen. This, together with the fact that the descriptions are usually incomplete or ambiguous, and that low-level categories are sometimes complex and full of exceptions, makes it a hard classification problem, even for domain expert humans.

3.2. Preprocessing

The first step was to preprocess the data, bringing it to a workable shape. For this purpose, several datasets provided by INE for the task were joined into a single cohesive file. Next, several preprocessing

[2] https://www.ine.es/daco/daco42/clasificaciones/cnae09/notasex_cnae_09.pdf (visited on May 2025).

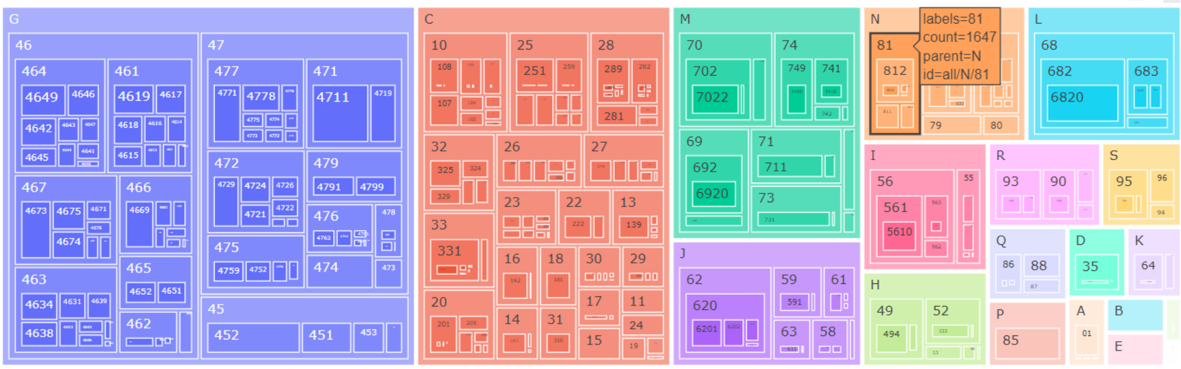

Figure 2: Treemap showing the hierarchical structure of the dataset. The numbers represent each different class, and the rectangle size is proportional to the amount of data in each category.

no. samples per class	no. classes	pct. classes
> 1,000	342	54.37%
> 5,000	125	19.87%
> 10,000	62	9.85%

Table 1
Sample distribution in the resulting working dataset.

and data augmentation techniques were applied to improve and increase the amount of data available. In order to homogenize the descriptions while keeping as much useful textual information as possible, we applied the following techniques:

- Converted all text into lowercase.
- Replaced all the accents and non-UTF characters by their UTF equivalents, barring the ñ character due to being very common in Spanish.
- Removed all numbers.
- Removed all punctuation.
- Removed line breaks and extra whitespaces.

Next, we build synthetic descriptions based on the titles and explanatory notes of each category of the CNAE-2009. We apply data augmentation to enlarge this subset by taking a specialized dictionary of synonyms (already used by INE for manual classification) and replacing the original words by their equivalents. This process allows us to increase diversity without affecting the meaning of the samples. By doing this, we obtained a training set of approximately 3.2 million instances, distributed as shown in Figure 2.

However, with CNAE-2009 containing 629 classes, the resulting dataset keeps being imbalanced, even after adding synthetic data: the most common class accounts for 10.1% of the instances while, in contrast, the least common accounts only for 0.00013%. As shown in Table 3, there is a big tail of minority classes, with half of them (45.63%) containing less than 1,000 samples. The main reason behind this severe imbalance is the structure of the Spanish economy itself, where some activities are much more common than others.

3.3. Evaluation methods

Due to the imbalance present in the dataset, we chose to use the *weighted F1* score as our evaluation metric to measure the performance of our classification models. This is an extension of traditional F1 score for multi-class classification problems, where traditional *F1 score* is the harmonic mean of two key metrics: *Precision* (fraction of true positive predictions out of all positive predictions made by the model) and *Recall* (fraction of true positive predictions out of all actual positive instances). Thus, the F1 score balances the trade-off between these two metrics.

Model	Accuracy	Wgt F1 Score
FastText	**0.9008**	**0.8913**
bert-base-uncased	0.5545	0.4784
roberta-base	0.6973	0.6416
PlanTL-GOB-ES/roberta-base-bne	0.6925	0.6265
PlanTL-GOB-ES/roberta-large-bne	**0.8695**	**0.8478**
PlanTL-GOB-ES/roberta-large-bne-massive	0.7835	0.7400

Table 2
Results obtained for each model.

4. Development

4.1. First Approach: FastText

Firstly, we implemented a FastText-based model, which will be used as our baseline. After a previous tuning of the metaparameters,[3] we obtained the results shown in Table 2 for the validation set. We also tried to initialize the weights of the model with Spanish pre-trained ones, but no improvement was obtained.

4.2. Second Approach: Transformers

After implementing the FastText-based classifier, a baseline model was now available for comparison. Next, we proceeded with our transformer-based approach by fine-tuning several BERT models that had been previously trained with Spanish texts:

- **bert-base-uncased** [5]: The first BERT model, pretrained on a large corpus of English data in a self-supervised fashion.
- **roberta-base** [6]: A BERT variant improved by training on larger data, removing next sentence prediction, and using larger batches. This resulted in better performance on language understanding benchmarks.
- **PlanTL-GOB-ES/roberta-base-bne**:[4] This Spanish language model is based on the RoBERTa base model and has been pre-trained using the largest Spanish corpus known to date. This corpus contains 570 GB of clean and deduplicated text processed for this work, and compiled from the web crawlings performed by the *National Library of Spain* from 2009 to 2019.
- **PlanTL-GOB-ES/roberta-large-bne**:[5] Based on the *roberta-large* architecture, and trained with the same data as *roberta-base-bne.*
- **PlanTL-GOB-ES/roberta-large-bne-massive**:[6] This is an Intent Classification model for the Spanish language, fine-tuned from a RoBERTa based model pre-trained on MASSIVE 1.1. This is a parallel dataset of more than 1M utterances across 52 languages, with annotations for the Natural Language Understanding tasks of intent prediction and slot annotation.

For the fine-tuning of these models, the dataset, containing more than 3.2 million samples, was split in two: 90% for training and 10% for validation. Taking into account the imbalanced nature of the data, we decided to use a *stratified cross-validation* with 5 folds. This way, we ensure that the model is trained and validated among all available examples present in the dataset. Furthermore, by averaging the results of the folds we obtain a better estimate of the performance of our model.

Table 2 presents the obtained for the validation dataset. As we can see, the *bert-base-uncased* severely underperformed, as it was expected, due to it being a smaller model trained in English texts.

[3] No. of epochs: 10. Learning rate: 0.1. Max. ngram length: 3.

[4] https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne (visited on May 2025).

[5] https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne (visited on May 2025).

[6] https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne-massive (visited on May 2025).

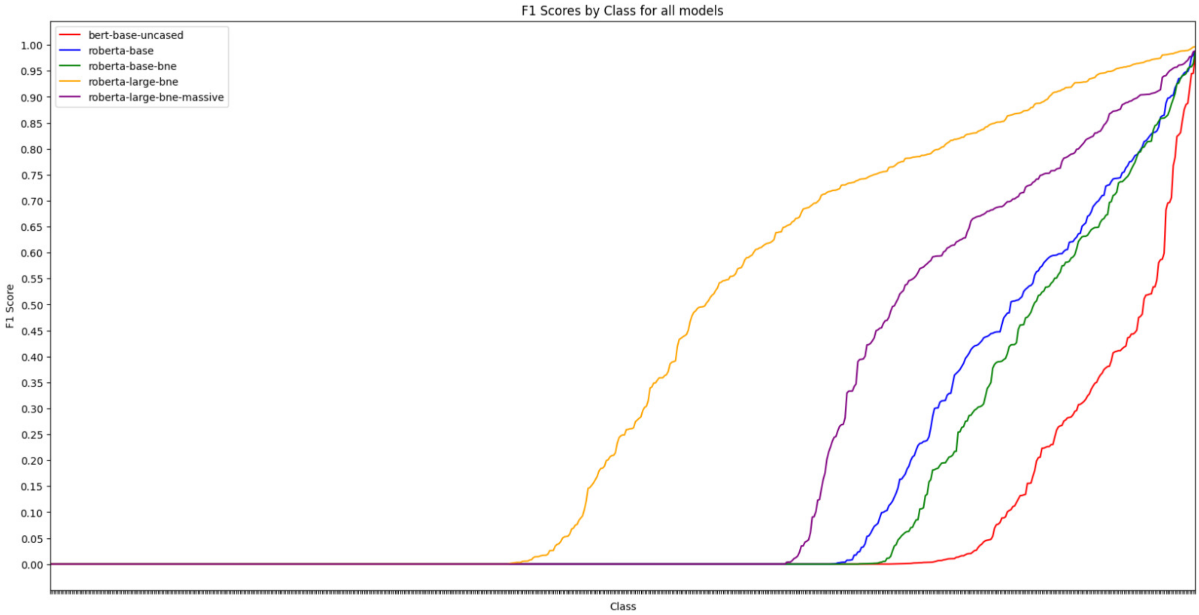

Figure 3: F1 scores per class achieved for each model, in ascending order. The x-axis contains all the 629 classes present in the dataset.

F1 Score	pct. classes
> 0	59.93%
> 0.50	42.76%
> 0.75	28.13%
> 0.90	12.87%

Table 3
F1 scores achieved by the *roberta-large-bne* model across the entire 629 classes.

Surprisingly, *roberta-base* slightly outperformed *roberta-base-bne*, despite being trained on English texts while *roberta-base-bne* was trained with Spanish ones. The best performance was obtained with *roberta-large-bne*, with a bigger architecture, even outperforming *roberta-large-bne-massive*, which shows that *roberta-large-bne-massive* generalizes worse due to being pretrained with a more specific dataset.

In Figure 3 we can see the F1 score per class obtained by each model. On the x-axis we plotted each class, and in the y-axis their F1 score. The classes are ordered by F1 score in each model, from lowest to highest. This is only meant to be a graphical representation of performance across all classes, not a strict comparison on how each class performed with each model.

Next, we chose to delve deeper into *roberta-large-bne*, as it had achieved the highest score. We studied the individual F1 score that obtained for each class, as shown in Table 3. A F1 score between 0 and 0.50 means that the classification is worse than random, and a F1 score of 0.50 is as good as a random guess. Higher scores, around 0.75, represent a decent classification, and those higher than 0.90 involve a really good classification. These results are related to the amount of samples present per class, as 287 classes have less than 1,000 samples, and out of those classes, only 54 achieved an F1 score higher than 0. Overall, the model fails to classify around 41% of all classes present in the CNAE, which is not a desirable result.

4.3. Third Approach: LLMs

Finally, we intented to try out API-based LLMs and some smaller local models that could be fine-tuned with our available dataset. Regarding the API-based implementation, we tried two strategies, both of them using the free version of ChatGPT:[7]

1. A *multi-class approach* using *zero-shot learning*. We asked the LLM to categorize a text according to CNAE-2009 without further information or instructions. This first naive strategy was unsuccessful, since the system was not able to classify at all.
2. A *multi-label approach* through hierarchical classification. The LLM was asked to provide the right section, then the division (choosing from the ones included in the given section), and so on. We added the titles of each level categories on the prompt, which could be considered *few-shot learning*. The performance was again unsatisfying as the system wasn't able to classify, even with the given instructions.

Notice that we could not include the whole explanatory notes of the possible categories in the prompt, since it exceeded the token limit for the free version of ChatGPT we were using. However, it is likely that, even with a paid version of any LLM, the prompts would not be able to fit all the information pertinent to each CNAE code.

In the case of local models, we have tried the following:

- **meta-llama/Llama-2-7b** [8]: *Llama 2* is an auto-regressive language model that uses an optimized transformer architecture. We chose the 7 billion parameter version, that supports a wide variety of languages.
- **mistralai/Mistral-7B-v0.1** [9]: Another pre-trained generative LLM with 7 billion parameters that allegeudly surpass *Llama 2*'s performance.
- **meta-llama/Llama-3.2-3B**: An updated version of *Llama* with a reduced size of parameters to fit in less powerful hardware.

Unfortunately, in spite of our attempts, none of these models fitted in our currently available GPUs. Thus, we could not fine-tune and test any of these models.

5. Future Research

As we expected due to the imbalanced distribution of samples in the dataset, our FastText model could outperform the more complex BERT-based approaches. Transformer-based models require more data to be properly trained; thus, we will need to perform improvements on the dataset or try a different classification approach. Currently, we are working in several points to improve upon these previous experiments.

5.1. Expanded datasets

Our first experiment would be to reduce the number of samples in the dominating classes, and find the threshold were their performance starts to decrease. This would make training take less time, and may allow the models to be able to focus more on the minority classes. Another improvement would be to generate entries to reduce the imbalance gap in the dataset. For this purpose, we have created the `base_filtered` dataset.

By making all classes reach at least 2,000–5,000 examples, we expect to improve the ability of the model to classify them. For this purpose, a prototype version of a data generation tool is in the works. By using LLMs via prompting, and giving them what information constitutes each class alongside some examples, they are able to produce working examples, albeit they are ”too perfect”. This generation method still needs to be improved upon and refined, for it to give more realistic examples. For now, we have managed to generate 100 samples per class.

- **`base_data` (3.24M samples)**: This is the dataset explained in detail in Section 3.1. It is severely imbalanced.

[7] https://chatgpt.com/

- **base_filtered (2.23M samples)**: Dataset created from base_data, where clustering was applied to reduce the sample size of the majority classes in an attempt to combat the imbalance in the results. Clustering was performed using SentenceTransformers (distiluse-base-multilingual-cased-v2 model), and the 10 majority classes were reduced to 20,000 elements each. However, a severe imbalance still exists, but less than with base_data.
- **generated_data (63K samples)**: Dataset generated with LLama 70B, containing 100 entries per CNAE class. Its entries consist exclusively of better-written sentences and are generally longer than the vast majority of real-life examples of CNAE descriptions provided by companies.
- **generated_data_2 (74K samples)**: A mixture of generated_data and a small dataset consisting of 10.000 entries proportioned by INE. By mixing these two data sets, the class balance present in generated_data was lost, but it is not as significantly unbalanced as in base_data.
- **codauto_data (3.30M samples)**: New version of base_data which features some more examples, but the data distribution is very similar to base_data. It is being used to train the production version of the Codauto classifier, an internal classifier that INE is developing.

INE also developed a test dataset in order to standardize the comparisons between models. This set is made of **1,654 samples** distributed among four categories:

- **confident_learning (747 samples)**: This subset has been developed using confidence learning algorithms on the base_data dataset to extract the most relevant and varied samples.
- **handwritten (31 samples)**: This subset contains very few entries that have been hand-written by experts at INE. They specifically target categories that have very similar entries, in order to determine if the models are able to classify samples with subtle differences.
- **queries (246 samples)**: This subset is made of real world examples of submitted company activity descriptions. This subset is the most important, as the models' performance over it reflect more closely their theoretical performance when deployed.
- **train_set (629 samples)**: This subset is made of examples from base_data, thus when the models are trained with either base_data or base_filtered, they should show solid results. The entries in this subset can be the result of data augmentation, so their quality isn't expected to be the best.

5.2. Further model tests

We executed the FastText and roberta model with the new datasets, as well as a new LLM: Salamandra-2B [10]. This time we were able to train it with better hardware, alongside some code optimizations. Table 4 shows the results of the multiple experiments we've performed.

As we can see, FastText still outperforms the rest of the models. Even though, our experiments show that the Transformer models perform well with the generated datasets, which upon further expanding could result in the best performing model. We will focus our efforts in improving our dataset generation tools, to create a more definitive set.

5.3. Future work

As future work, we intend to expand the Transformer-based classifiers as we think its the approach that has the most potential. The two main approaches we plan on working on are:

- A hierarchical classifier, that will take advantage of the formatting of the CNAE codes. As the codes are already hierarchical, this could allow for partial classifications following the Section, Division, Group and lastly Class of every code.
- A ranked classifier with which we would obtain the top k most relevant classifications for a given example which given the nature of the problem could result in a more lenient system.

Finally, as the results of our LLM-based approach were underwhelming we are interested in testing more in depth these models. We need to get more familiarized with the literature on this topic, to be able to choose the future approaches to take.

Model	Dataset	Conf Learn		Handwritten		Queries		Train_set	
		Acc	wF1	Acc	wF1	Acc	wF1	Acc	wF1
FastText	Base	0.63	0.63	0.52	0.51	0.46	0.49	0.72	0.73
	Filtered	0.60	0.60	0.49	0.48	0.47	0.45	0.67	0.66
	Generated	0.26	0.23	0.22	0.22	0.24	0.25	0.42	0.41
	Generated_2	0.29	0.28	0.26	0.27	0.25	0.26	0.46	0.45
	Codauto	**0.66**	**0.66**	**0.55**	**0.52**	**0.49**	**0.50**	**0.75**	**0.74**
PlanTL-GOB-ES/roberta-large-bne	Base	0.46	0.42	0.48	0.41	0.35	0.34	0.61	0.58
	Filtered	**0.49**	**0.47**	**0.52**	**0.44**	0.33	0.30	**0.65**	**0.63**
	Generated	0.21	0.21	0.19	0.15	0.30	0.29	0.31	0.32
	Generated_2	0.36	0.32	0.32	0.32	**0.38**	**0.37**	0.51	0.48
	Codauto	0.48	0.44	0.39	0.30	0.32	0.27	0.60	0.57
Salamandra-2B	Generated_2	0.25	0.21	0.28	0.23	**0.24**	**0.22**	0.26	0.23
	Codauto	**0.33**	**0.31**	**0.42**	**0.43**	**0.24**	**0.22**	**0.50**	**0.48**

Table 4
Performance of FastText, roberta-large-bne and Salamandra-2B models using the various available datsets. Metrics used are Accuracy and Weighted F1 Score

Acknowledgments

CIDMEFEO funded by the Spanish National Statistics Institute (INE); as well as funding by Xunta de Galicia (ED431C 2024/02), and Centro de Investigación de Galicia "CITIC", funded by the *Xunta de Galicia* through the collaboration agreement between the *Consellería de Cultura, Educación, Formación Profesional e Universidades* and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).

CITIC, as a center accredited for excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Department of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. Additionally, it is co-financed by the EU through the FEDER Galicia 2021-27 operational program (Ref.ED431G 2023/01)

Declaration on Generative AI

We have used ChatGPT for minor copy-editing and Grammarly for grammar and spelling check. After using these tools, we have reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. M. Kibriya, E. Frank, B. Pfahringer, G. Holmes, Multinomial naive Bayes for text categorization revisited, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2004, pp. 488–499.

[2] T. Joachims, Text categorization with Support Vector Machines: Learning with many relevant features, in: C. Nédellec, C. Rouveirol (Eds.), Machine Learning: ECML-98, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 137–142.

[3] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1986) 81–106.

[4] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146. doi:10.1162/tacl_a_00051.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[7] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, OpenAI Blog (2018). URL: https://openai.com/index/language-unsupervised/.

[8] H. Touvron, L. Martin, K. Stone, Llama 2: Open foundation and fine-tuned chat models, 2023. URL: https://arxiv.org/abs/2307.09288. arXiv:2307.09288.

[9] A. Q. Jiang, A. Sablayrolles, A. Mensch, Mistral 7B, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[10] A. Gonzalez-Agirre, M. Pàmies, J. Llop, I. Baucells, S. D. Dalt, D. Tamayo, J. J. Saiz, F. Espuña, J. Prats, J. Aula-Blasco, M. Mina, A. Rubio, A. Shvets, A. Sallés, I. Lacunza, I. Pikabea, J. Palomar, J. Falcão, L. Tormo, L. Vasquez-Reina, M. Marimon, V. Ruíz-Fernández, M. Villegas, Salamandra technical report, 2025. URL: https://arxiv.org/abs/2502.08489. arXiv:2502.08489.