Language Models to fill-in-the-blanks

María Viana Rozas
Universidad del País Vasco/
Euskal Herriko Unibertsitatea
mviana009@ikasle.ehu.eus

Abstract

The following paper consists on a review of the document presented by Chris Donahue et al., in which they show a new approach to the task of fill in the blanks. In addition, as a personal contribution and based on their Mask Function I will show how I have made my two models (one for Spanish, and one for German) that can be used for fill-in-the-gaps.

1 Introduction

Chris Donahue et al. (Donahue et al., 2020) present a new approach to the fill in the gaps task. Their purpose is to extend the resources that are usually used for this task making use of their own Language Model (LM) to prove that it gives a broader performance than others, allowing masking at different syntactic levels (n-grams, sentences and paragraphs). To do so, they perform a new Mask function and apply four different methods to finally demonstrate that, compared to other models such as Bert or SA, their ILM model has a better performance, since it carries out a more accurate infilling (based on Human Evaluation).

2 The Infilling Task

The task of text infilling consists on filling missing gaps of a sentence or a paragraph. As Wanrong Zhu et al. (Zhu et al., 2019) explain, their use can be related to restoration of historical or damaged documents, contract or article writing or text editing, among others.

The problem with this type of task is related to find the model that performs better when doing the masking. Wanrong Zhu et al. (Zhu et al., 2019) argue that the most appropriate model for this type of task is a self-attention mechanism, since it allows the left- and right-side context to be taken into account.

In this case, Chris Donahue et al. (Donahue et al., 2020) propose a new approach to the infilling task.

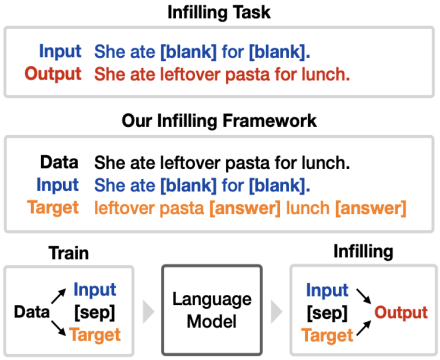

Figure 1: LM for the infilling task (Donahue et al., 2020)

Its purpose is focused on the generation of stories from a context. Moreover, as there are currently models dedicated to this type of task that give a good performance, such as T5, GPT and BART, the authors of the paper make use of GPT-2 'small' in their training.

As shown in the figure (See figure 1), they make use of a LM to predict missing words. Their LM masks random spans and generates pairs of inputs and targets (where they keep the answers, being these the words that were masked before).

3 Methodology

3.1 The mask function

The authors create a 'mask function', which can be customized to fill in different levels of text, such as individual words, phrases, sentences, or even entire paragraphs or documents. The mask function works by replacing the missing text with a special token that represents the level of granularity being filled in, such as a '[blank]' for individual words.

The mask function is configured to randomly mask around 15% of the tokens in a document, but instead of uniformly masking each token, it masks entire subtrees of the hierarchy tree with a 3% probability. This results in a marginal token mask rate of about 15% for the datasets being used.

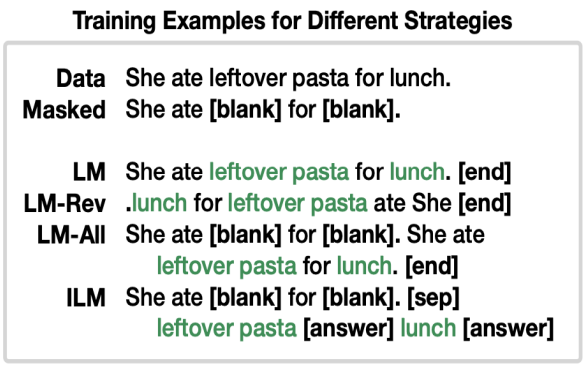

Figure 2: The four different approaches using training examples (Donahue et al., 2020)

Although the framework can fill in different granularities of text, the authors primarily evaluate the model's ability to fill in missing sentences.

3.2 The Datasets

They used three different datasets, so that their model was not set to only one exclusive domain:

- Stories: contains about 100K examples. Based on short stories from the ROCStories dataset.
- Abstracts: contains about 200K examples. Based on abstracts from CS papers on arXiv.
- Lyrics: contains about 2M examples. Based on song lyrics from lyrics.com.

3.3 The Methodology

They consider four different approaches to check the one that performs better, using GPT-2 'small', using the same hyperparameters and changing the infilling strategy and dataset:

- Language Model (LM): using as context only what's before the blank spaces.
- Reverse LM: using as context only what's after the blank spaces.
- All-LM: using all available context.
- Iterative Language Model (ILM): on a given artificially-masked sentence.

As shown in the figure 2, that is how the four methodologies are used to fill in the blanks.

4 Training

The ILM framework first generates examples by randomly replacing some parts of a complete text example with '[blank]' tokens. These replaced parts are then concatenated with '[answer]' tokens to form a training target. The complete infilling example is formed by combining the original text example, the replaced parts, and the training target.

The framework then trains a language model on these infilling examples using standard language model training methodology. The trained model can predict the missing parts of a text example given the original text and the replaced parts.

This framework has several advantages. It's computationally efficient, as it only requires a small number of additional tokens to the original text example. It also requires minimal changes to the LM's vocabulary. Additionally, it allows the model to incorporate context from both sides of a blank, while still being able to decode from LMs.

5 Evaluation

5.1 Quantitative

To test the different approaches, they compare their performance on the different datasets based on perplexity (PPL).

Perplexity measures how well a LM is able to predict the next word in a sequence, so it's a good measure for this type of task. A low perplexity indicates that the model is good at predicting the next word, while a high perplexity indicates that the model is not so good at predicting the next word.

The results (Figure 3) show that ILM is the best performing model, although LM-All could be one of the best when taking into account the whole context. As the authors point out, this demonstrates that GPT-2 'small' 'is able to effectively learn the 'syntax' of ILM examples and achieve reasonable infilling performances with shorter sequences'((Donahue et al., 2020)).

	STO	ABS	LYR	Length
LM	18.3	27.9	27.7	1.00
LM-Rev	27.1	46.5	34.3	1.00
LM-All	15.6	22.3	21.4	1.81
ILM	15.6	22.4	22.6	1.01

Figure 3: Quantitative Results (Donahue et al., 2020)

5.2 Qualitative

They also used human evaluation to check the performance of their models compare to others like SA (self-attention) and BERT. In order to check the performance, they generate a story from the stories dataset and randomly replace one of its five human-written sentences with a model output.

They fine-tune the models with the stories dataset. Finally, for the comparison, they give a score based on the percentage of examples where the annotators did not identify the machine-generated sentence. ILM has the best score, proving that it's the most accurate for the task:

	BERT	SA	LM	ILM
Score (%)	20	29	41	45

Figure 4: Qualitative Results (Donahue et al., 2020)

6 My personal Approach

6.1 Introduction

My approach consists on performing a fine-tuning of the bert-base-spanish-wwm-cased bert with a dataset of Tweets from the economic-political domain, and a fine-tuning of the distilbert-base-german-cased with a dataset of Amazon Reviews. The intention is to use this models combined with the idea of Chris Donahue et al. (Donahue et al., 2020) presented before. Through my models, that use [MASK] tokens to predict the missing words, I use their main idea adapted to the possibilities of my models. [1]

6.2 Bert

BERT stands for 'Bidirectional Encoder Representations from Transformers'. It is a pre-training technique for natural language processing (NLP) developed by Google AI Language in 2018. BERT uses Masked Language Model (MLM). The MLM randomly masks some tokens from the input, and the model is trained to predict the original vocabulary id of the masked word based only on its context. Unlike left-to-right language model pre-training, the MLM objective enables the representation to fuse the left and the right context, allowing pre-training of a deep bidirectional Transformer, which is how the authors used their model.

I chose to work with a BERT available in the Hugging Face repository because it gives good results in NLP tasks and it's also relatively easy to fine-tune. For my particular case I chose the 'bert-base-spanish-wwm-cased', but it can also be used the uncased version.[2] This spanish 'BETO' is based on the same architecture as the original BERT model, including 12 transformer layers with multi-head self-attention.

The model's vocabulary is based on a large corpus of Spanish language text and includes over 100,000 subword tokens. Like the original BERT model, it uses the Masked Language Model (MLM), but with the addition of Whole Word Masking (WWM), which helps when preserving the meaning of the words.

The model can be fine-tuned for a variety of Spanish language processing tasks, such as text classification, sentiment analysis, and named entity recognition. So what I've done is fine-tuning the dataset of Tweets [3] related to the economic-political sphere so that I can then use it for the infilling task.

Regarding the dataset, it has not been annotated in any way, which is not important for the task in which it is going to be used. Furthermore, the content is divided into rows and for each row a Tweet that is commented as in favor or against the Colombian president Gustavo Petro.

6.3 DistilBert

DistilBERT is a compressed version of the BERT model, designed to be smaller and faster, while still maintaining high performance on natural language processing tasks. DistilBERT has several advantages over the original BERT model. Firstly, as it is smaller and faster, it makes it more suitable for use in resource-constrained environments or on low-power devices. It also requires less training data, which can be a significant advantage when working with smaller datasets.

For this model I have decided to change the language, but there is also Distilbert for Spanish available in the Hugging Face repository. I chose distilbert-base-german-cased [4], which is a smaller version of the original German Bert (bert-base-german-cased), and it has 83 million parameters. It

[1] All the code is available in the colab Notebook I created (Viana)

[2] Available in Hugging Face: https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

[3] Available in Hugging Face: https://huggingface.co/datasets/jhonparra18/petro-tweets

[4] Available in Hugging Face: https://huggingface.co/distilbert-base-german-cased

was pretrained on a large corpus of German text, using a masked language modeling (MLM) objective.

I performed the same fine-tuning process as before, but using in this case a new dataset: Amazon-reviews-multilingual in German [5], to extend the distilbert domain to a product sales related approach. Regarding the dataset, it's a large, multilingual dataset of customer reviews of products sold on Amazon. It contains over 200 million reviews across multiple languages, and includes both binary and multi-class labels for sentiment analysis. The dataset has also been preprocessed to remove noise and ensure data quality.

6.4 Training

For the training part, I did the same for both models using the Hugging Face Transformers library: I start by downsampling the original training dataset to train_size and splitting it into training and validation sets with the ratio specified by test_size. Then, I specify the arguments for the Trainer, including the output directory, evaluation strategy, learning rate, weight decay, batch size, and number of training epochs (which are different for both models, but between 10 and 20 is enough, because it takes a lot of time).

The results for these Training for both models can be seeing in the table 1. Note that this is not a comparison of the models, because they were used with different languages, so they should only be compared with other models using the same language:

Models	Training loss	Validation loss
Spanish-BERT	2.2313	2.7373
German-Distilbert	3.944	3.8668

Table 1: The training results show that Distilbert has more loss than the spanish-Bert. This can be due to the difference between the models, but also because of the different number of epochs used during training and the size of the dataset used.

6.5 Evaluation

For the evaluation part I have chosen to determine the Perplexity, since it is a widely used metric to determine the performance of LMs. Although the Perplexity of the Bert is better than that of the DistilBert (See Table 2), they are not really comparable, since they use different languages, but it's still worth it to check their perplexity results.

I consider that it would not be fair to compare these results with GPT2-small and other Spanish or German models such as Roberta or Bart, since they are much larger models, with more parameters and also more focused on text generation and not on the infilling task, which although related in a way, are not exactly the same thing.

Models	Perplexity
Spanish-BERT	15.17
German-Distilbert	48.78

Table 2: Resulting Perplexity for both LMs.

6.6 ILM in my own models

In this part what I do is adapt the main idea of the previously mentioned paper from Chris Donahue et al., because their ILM is not directly applicable to my own models. The basic idea behind it is to create my own fill-in-the-blank prompts using my own tokenizer and model. I create a context (such as 'La ___ es azul') related to the domain of my dataset, tokenize it using my own tokenizer, replace the blank (s) with special token(s) that my model recognizes as placeholders, and then pass the resulting tokens through my model to generate predictions for the missing word (s).

For this part I've considered two different approaches for the two models:

- Generating only one [MASK] in the sentence.
- Generating two [MASK] in the sentence.

For the first one I used context and left this option out for the second one, to check which of the two options gives better results. As the results show, it is easier for the LM to give better predictions when given a context from which it can learn.

As can be seen in Figure 5 and Figure 6, context plays an important part in improving the results. These sentences are just an example of the two methods performed with Spanish-Bert, but more examples of sentences can be found in appendix A.

In addition, it is necessary to emphasize that in the examples in the appendix it can be seen how some German sentences have spelling mistakes, related to the conjugation of the verb. This can be due to the fact that there are no Word Embeddings in the DistilBert, which affects the quality of the performance.

[5] Available in Hugging Face: https://huggingface.co/datasets/amazon_reviews_multi

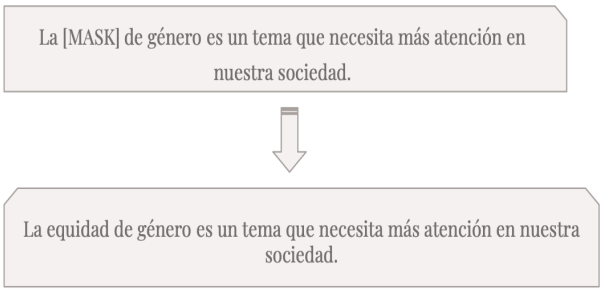

Figure 5: Sentence with one [MASK] and context ('La corrupción es uno de los principales problemas en la política y la sociedad actual.')

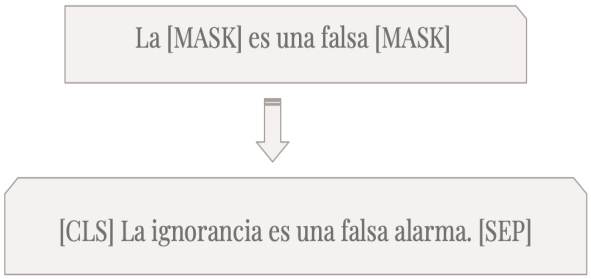

Figure 6: Sentence with two [MASK] without context

7 Conclusions

This new approach allows to advance in the infilling task, since it not only has a masking that takes into account the context from left to right, but also includes the possibility of replacing missing tokens by words, n-grams, sentences and even paragraphs, which is a breakthrough in infilling. In addition, this new approach could also be used for story development and text completion.

Regarding my own development, it is necessary to emphasize that this is a first approximation to the work done by Chris Donahue et al. (Donahue et al., 2020) It needs further study to really turn its masked function into a version available for different languages and/or domains.

8 Further work

Regarding the work presented by Chris Donahue et al. (Donahue et al., 2020) I would recommend broadening the domains in which ILM works, since I believe that it may be worthwhile to use this new strategy for the infilling task. In addition, I would advise expanding the languages used, so that this new approach can be exploited by more linguistic communities.

As for my own work, in the future it would be necessary to change the datasets, making use of some that were more accurate for the infilling task, and it would also be necessary to compare these models with two others, for example, a Spanish Distilbert and a German Bert to compare which type of model works better (keeping the same dataset in the 4 models, just changing the language).

References

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. *PML4DC, ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv (Cornell University)*.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *arXiv (Cornell University)*.

William Fedus, Ian Goodfellow, and Andrew M.Dai. 2018. Maskgan: better text generation via filling in the _. *arXiv (Cornell University)*, pages 2–9.

Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquín Silveira-Ocampo, Casimiro Pio Carrino, Carme Armentano-Oller, Carlos Rodriguez-Penagos, Aitor Gonzalez-Agirre, and Marta Villegas. 2022. Maria: Spanish language models. *arXiv*.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv (Cornell University)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv: Computation and Language*.

María Viana. Colab notebook with all the code.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do rnn language models learn about filler–gap dependencies? *Empirical Methods in Natural Language Processing*.

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *arXiv (Cornell University)*.

A Appendix A

A.1 Sentences with one mask in the spanish-Bert

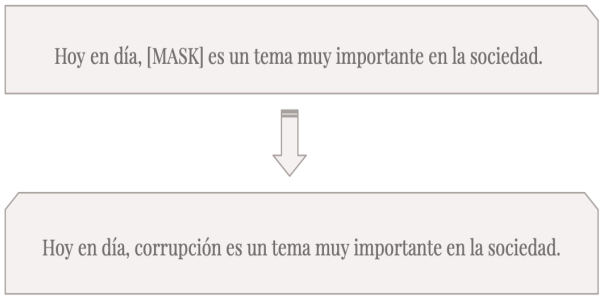

Figure 7

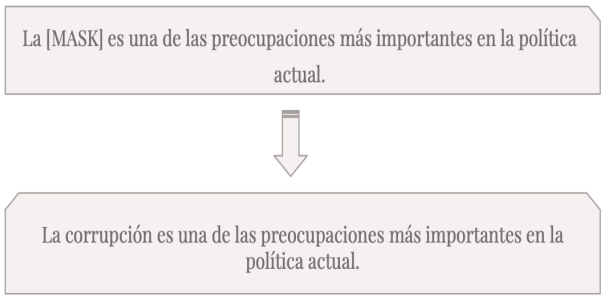

Figure 8

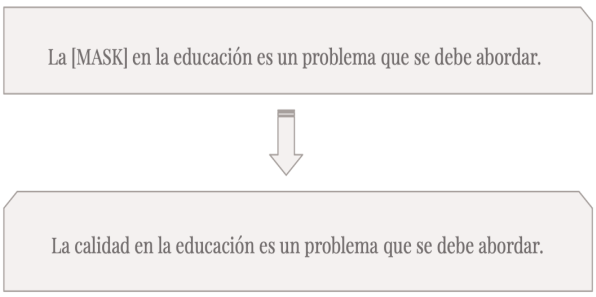

Figure 9

A.2 Sentences with two mask in the spanish-Bert

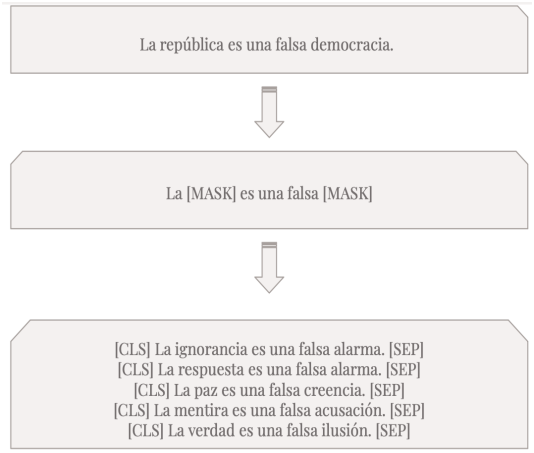

Figure 10

A.3 Sentences with one mask in the german-Distilbert

Figure 11

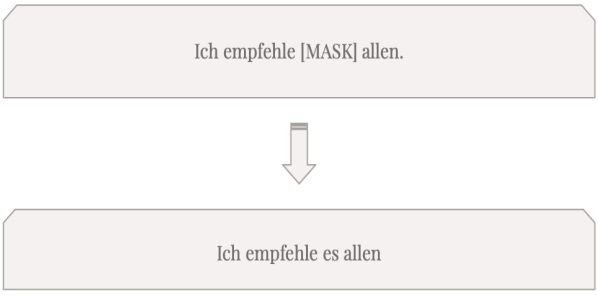

Figure 12

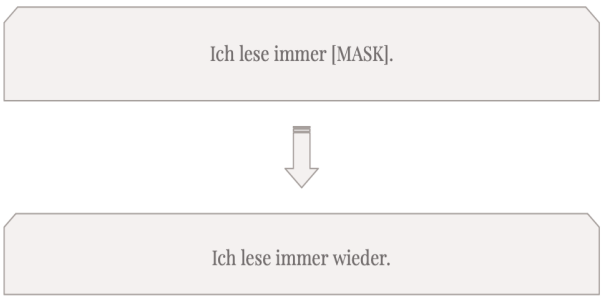

Figure 13

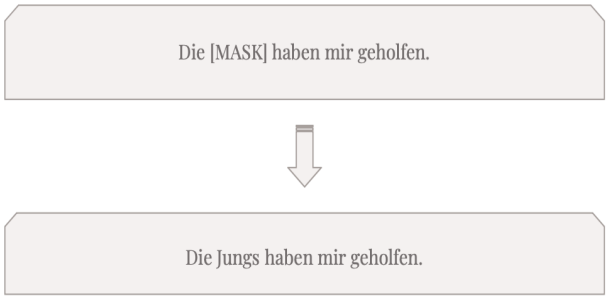

Figure 14

A.4 Sentences with two mask in the german-Distilbert

Figure 15