Procesamiento del Lenguaje Natural, Revista nº 67, septiembre de 2021, pp. 59-68 recibido 07-05-2021 revisado 01-06-2021 aceptado 06-06-2021

AutoPunct: A BERT-based Automatic Punctuation and Capitalisation System for Spanish and Basque

AutoPunct: Sistema de Puntuación y Mayusculización Automático basado en BERT para Castellano y Euskera

Ander González-Docasal[1,2], Aitor García-Pablos[1], Haritz Arzelus[1], Aitor Álvarez[1]
[1]Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastián (Spain)
[2]University of Zaragoza, Pedro Cerbuna 12, 50009 Zaragoza (Spain)
{agonzalezd, agarciap, harzelus, aalvarez}@vicomtech.org

Abstract: The raw output of an Automatic Speech Recognition system usually consists in a stream of words without any casing nor punctuation. In order to improve the readability and enable further uses of this output, punctuation and capitalisation have to be included. In this context, we present AutoPunct, a Transformers-based automatic punctuation and capitalisation model that combines both acoustic (i.e. silences duration) and lexical information (the words themselves). We compared its performance with a system based on Bidirectional Recurrent Neural Networks (BRNN) on Basque (a low-resource language) and Spanish, both individually and simultaneously. The result is a system that achieves high accuracy for punctuation and capitalisation in both languages at the same time, with a throughput of several thousand words per second using a standard GPU.
Keywords: punctuation, capitalisation, low-resource languages.

Resumen: La salida en bruto de un sistema de Reconocimiento Automático del Habla generalmente consiste en una secuencia de palabras sin mayúsculas ni signos de puntuación. Para mejorar la legibilidad y posibilitar posteriores usos de esta salida es necesario incluir la puntuación y las mayúsculas. En este contexto, presentamos AutoPunct, un modelo para puntuación y mayusculización basado en arquitecturas de Transformers que combina tanto información acústica (silencios) como léxica (palabras). Hemos comparado su desempeño con un sistema basado en redes neuronales recursivas bidireccionales (BRNN) en euskera (un idioma de pocos recursos) y castellano, así como combinando ambos idiomas. El resultado es un sistema que obtiene buenos resultados aplicando mayusculización y puntuación de manera simultánea en dos idiomas diferentes, con una velocidad de proceso que alcanza varios miles de palabras por segundo en una GPU estándar.
Palabras clave: puntuación, mayusculización, lenguas con pocos recursos.

1 Introduction

Automatic Speech Recognition (ASR) systems are increasingly more integrated in our daily lives and workflows through different solutions such as voice assistants, natural interfaces, speech-to-text applications or biometrics, among others. The growth of this technology has been mainly due to the evolution of Deep Learning techniques and their integration to develop neural models for speech recognition (Nassif et al., 2019), in addition to the continuous release of more and more training data and the availability of powerful hardware devices for high performance computing.

The aim of an ASR system is to transform an audio input into text that may be exploited for further Natural Language Processing applications. However, the output string is usually composed by a raw sequence of words which does not include casing nor punctuation marks, which noticeably reduces its readability (Jones et al., 2003) and its possibility of being employed as input to other modules that require a well-segmented and correctly punctuated text (Peitz et al., 2011). Therefore, the ASR module is usually concatenated to other technological modules which

ISSN 1135-5948. DOI 10.26342/2021-67-5

are in charge of enriching the initial raw transcriptions, as it is described in Figure 1.

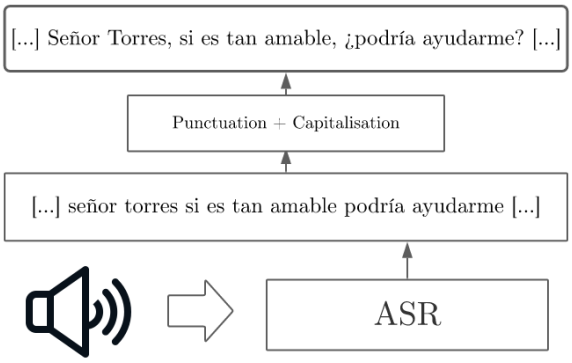

Figure 1: Example of Automatic Speech Recognition output before and after adding punctuation and capitalisation.

In this work, we compared the performance of two neural architectures for the task of automatically recovering the punctuation marks for Spanish and Basque languages, both individually and simultaneously. The first system was considered as a reference baseline. It is inspired on the architecture proposed by Tilk and Alumäe (2016), which includes a bidirectional recurrent neural network (BRNN) model (Schuster and Paliwal, 1997) that takes advantage of Gated Recurrent Units (GRU) as recurrent layers and an attention mechanism to further increase its capacity of finding relevant parts of the context for punctuation decisions.

As an alternative to this initial system, we present AutoPunct, which is composed by an architecture based on Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), that combines the text representation obtained by a Transformer model with the acoustic information related to the duration of the silences between words. We tested this second system using several architectural variations in order to evaluate different ways of combining and exploiting the acoustic and lexical information. In addition, this model variations were also evaluated with the integration of an extra head to predict whether a word should be capitalised or not.

The rest of the paper is structured as it follows. Section 2 introduces related work in the field. Section 3 presents AutoPunct along with its main architecture. Section 4 illustrates the initial experimental set-up, whilst the datasets used to train both neural approaches are described in Section 5. Section 6 displays the experiments and the obtained results. Finally, Section 7 concludes the paper and presents future work.

2 Related work

The challenge of automatically recovering capitalisation and punctuation marks has been extensively studied through many systems presented in the literature. These systems can be divided into three main categories (Yi et al., 2020a): those using lexical features, derived from the text; those using prosodic features, derived from acoustic information; and those using a combination of both. These prosodic feature-based architectures show that this type of information is indeed useful for the task, although they tend to fail in places where the speaker does unnatural pauses (Christensen, Gotoh, and Renals, 2001; Kim and Woodland, 2003).

In the last years, the problem of recovering punctuation marks have been faced by the use of Deep Learning algorithms, such as Convolutional Networks (Che et al., 2016), Bidirectional RNN with attention (Tilk and Alumäe, 2016), the use of word- and speech-embeddings (Yi and Tao, 2019), and more recently, Transformers based on BERT-like architectures (Devlin et al., 2019), which have been shown to obtain values as high as 83.9% (Courtland, Faulkner, and McElvain, 2020) on F_1 score in the well-known and reference IWSLT 2012 data set (Federico et al., 2012). Different architectures show the use of a pre-trained BERT model such as RoBERTa (Liu et al., 2019) in order to obtain the word-embeddings, which are fed to various networks based on BiLSTM (Alam, Khan, and Alam, 2020) or focal loss (Yi et al., 2020b), or aggregated across overlapping context windows for each individual token (Courtland, Faulkner, and McElvain, 2020). Another BERT-based architecture that performs both punctuation and capitalisation simultaneously can be found in (Sunkara et al., 2020), where the word- or subword-embeddings obtained from a pre-trained or a custom BERT are fed to two Softmax layers for punctuation and capitalisation respectively. These experiments, however, are focused on the specific domain of medical texts.

Nevertheless, the variety of literature dedicated to solve the problem of punctuation focuses only on the following three marks:

PERIOD (.), COMMA (,) and QUESTION (?). In addition, these results have been mostly evaluated over a single language: English.

The system presented in this paper addresses 13 different punctuation marks, which are described in the section 5. Moreover, the architecture of the proposed model features a multi-label output, which means that it can predict more than a single punctuation mark per word.

3 Description of the systems

As it was initially introduced, this work presents a comparison between two neural architectures for capitalisation and punctuation restoration for Spanish and Basque, both individually and simultaneously. These systems are described in more detail in the following subsections.

3.1 BRNN-based system

The architecture proposed for the first system is based on Punctuator (Tilk and Alumäe, 2016). It integrates a Bidirectional Recurrent Neural Network (BRNN) with an attention mechanism for restoring punctuation marks in unsegmented transcribed speech. This architecture allows the use of both GRU or LSTM layers as recurrent layers, whilst the attention mechanism increases the capacity of the model of detecting relevant context segments to improve punctuation decisions. Finally, the recurrent layers and the attention mechanism are coupled by a late fusion approach, which allows the output of the attention model to directly interact with the state of the recurrent layer while not interfering with its memory. A simplified diagram of this architecture is shown in Figure 2.

For this work, we trained the model in two different stages. In the first stage, only annotated text is used to train an initial model, so it learns to restore punctuation marks based on textual features only. In the optional second stage, a new model is estimated adding acoustic information as input, therefore learning to combine pause durations between words with textual features. The pause durations are obtained by using the start and end time-codes given by the speech recognition system at word level.

3.2 BERT-based system

Our BERT-based system, called AUTOPUNCT, combines the lexical information obtained from a pre-trained BERT model with the acoustic information coming from silence duration between words. These two sources of information are further combined using an additional Transformer model (a custom BERT trained from scratch). This system is trained to predict both punctuation and capitalisation at the same time. A high-level diagram of the architecture of AUTOPUNCT is shown in Figure 3.

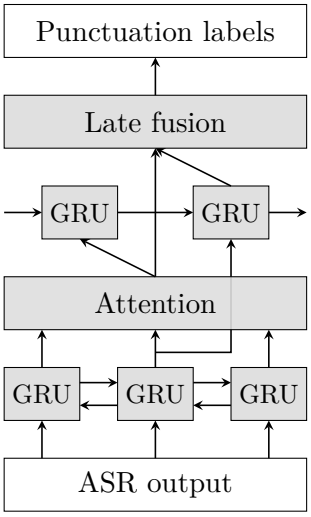

Figure 2: Simplified architecture of BRNN-based Punctuator system.

This particular architecture was carefully constructed considering the following guidelines:

1. Speed and efficiency should be prioritised so the system can work in real-time if needed.
2. Capitalisation should be performed alongside punctuation to avoid adding a standalone module just for casing.
3. The system should be able to exploit silence duration between words as an additional source of prosodic information.
4. The architecture should be language-independent. It should serve as a basis to train models for different languages by changing the training data and the pre-trained Transformer model.

In the following subsections, the main components of the AUTOPUNCT architecture are explained in more detail.

3.2.1 Input embeddings

The input to the AUTOPUNCT system corresponds to the output of a given ASR sys-

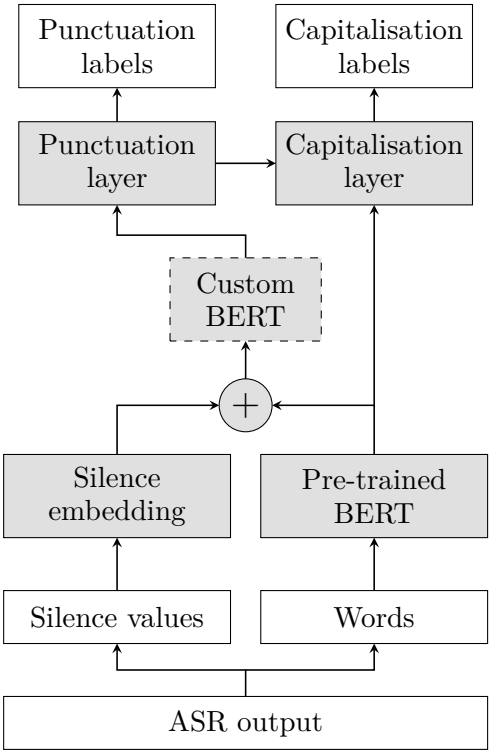

Figure 3: Main architecture of the BERT-based AutoPunct system.

tem; this is, a sequence of lowercase words paired with their silence duration value. A pre-trained BERT (Devlin et al., 2019) model is used to encode words as they are output by the speech recognition system into the so-called contextual embeddings. These embeddings are vectors of continuous values that encode relevant linguistic information about the words, according to the language model represented by BERT. More information about the pre-trained models used in the experiments can be found in Section 6.

This lexical and semantic information obtained by the pre-trained BERT is then combined with the silence duration for each individual word. Given that the silence values between word are scalars, they cannot be combined with the word-embeddings in this state. In order to solve this problem, we evaluated two different approaches of injecting the silences information to the system: as continuous values and as discrete values. The former involves repeating the silence value as many times as the size of each word embedding (768 for the BERT-base models). The latter consists of partitioning a ten-seconds-time range into buckets of 10 milliseconds, leading to 1000 discrete silence values. These discrete values are used as indexes over a silence-embedding layer that provides, for each silence value, a trainable vector of the same size as the word-embeddings.

Nevertheless, using discrete silence values with such a fine granularity has a major drawback: most of the discrete values will probably never appear in the training data; therefore their embeddings will not be used and they will not capture any valuable information. To prevent this, all the embeddings from a window of 1 second centred on the original value are averaged into a single embedding. We evaluated the system with two types of averaging: uniform averaging, in which each embedding weights the same, and Gaussian averaging, in which a Gaussian distribution centred on the original embedding was used to compute a weighted average of the embeddings. This final silence embedding is added to the word-embedding computed by the pre-trained BERT in order to obtain a combined representation.

3.2.2 Custom BERT

As an additional step, we evaluated the system adding an extra Transformer module layer after the word and silence combination. This custom Transformer, (henceforth Custom BERT) is much smaller than a base BERT (4 hidden layers with 4 attention heads each), and it is initialised from scratch. The rationale for adding this intermediate Transformer is to endow the model with the ability to attend to the whole sequence, instead of focusing on isolated combinations of word and silence pairs, through the self-attention among inputs to the Transformer.

The impact of this layer is also evaluated in Section 6.

3.2.3 Punctuation and Capitalisation layers

Finally, the representation of each word is fed into two classification heads. These heads are composed of a dense layer, followed by a non-linearity, a dropout layer and a final linear layer that maps the input into the corresponding label space, namely, the different punctuation marks for the punctuation head and the capitalisation labels for the capitalisation head.

Since the punctuation marks in a sentence may influence the capitalisation of the fol-

lowing words, the output of the punctuation head is fed into the capitalisation head along with the word embedding generated by the pre-trained BERT at the beginning of the network.

The punctuation head is treated as a multi-label classification head to cope with the fact that some words may bear more than one punctuation mark attached, e.g. closing quotation mark and period at the end of a sentence.

4 Training setup

Each neural architecture was initialised using specific components and resources, while the final models were constructed over the same main dataset employed in this work.

With regard to the BRNN-based system, we followed the training stages described in subsection 3.1. At a first stage, an initial model was estimated using text features only. These features were obtained from a text corpus of generic domain consisting of web news crawled from digital newspapers in Basque and Spanish. This generic text corpus is described in more detail in subsection 5.1. Then, the initial model was fine-tuned in a second stage of training with acoustic information, exploiting the principal dataset employed in this work, known as mintzai-ST (Etchegoyhen et al., 2021). This corpus is also described in more detail in subsection 5.2.

The models for all the languages were estimated using the same training configuration. During the first stage, the training finished when the validation perplexity was not improved at the first time, with a maximum epochs of 50 and a minibatch size of 128. The hidden layers consist of 256 units and we employed a learning rate of 0.02. Regarding the second stage where the acoustic information was integrated, the training process was set to be finished when the validation perplexity was not improved in the last 3 epochs with a maximum epochs of 50 and a minibatch size of 128. The hidden layers consist of 256 units and the learning rate was fixed to 0.02. The input vocabularies had a maximum size of 100,000 words, composed by the most frequent words that occur at least two times in the training corpus. The output vocabulary was composed by the predicted punctuation marks in addition to a non-punctuation symbol defined as `O`. The trainings were perfomed on a 12 GB Nvidia Titan X GPU card.

Regarding the BERT-based system AutoPunct, we employed a specific pre-trained BERT model for each language: BETO for Spanish (Cañete et al., 2020), BERTeus for Basque (Agerri et al., 2020), and IXAmBERT for Spanish+Basque (Otegi et al., 2020). As in the previous architecture, the final models of the BERT-based systems were estimated on the mintzai-ST dataset and using the same hyper-parameters to make them comparable. The learning-rate was set to $2 \cdot 10^{-5}$ with a warm-up period of 5 epochs and using AdamW (Loshchilov and Hutter, 2019) as the optimizer. The training mini-batches were of size 8. The training of each model was performed using an Nvidia GeForce RTX 2080ti GPU with ∼11GB of memory for a maximum of 50 epochs with an early-stopping patience of 20 epochs, monitoring the punctuation F_1 metric on the corresponding development set.

5 Main datasets

In this section, we first describe the text corpus used to estimate the initial models of the BRNN-based models, and then we present the main dataset employed to generate the final models of both neural architectures. Finally, we detail the punctuation and capitalisation labels as well.

5.1 Generic text corpus

This corpus is composed by news of generic domain crawled from digital newspapers from 2012 to 2019. The number of words for each partition is shown in Table 1.

	EU	ES	ES+EU
train	14,010,067	11,609,170	25,619,237
dev	683,230	1,323,545	2,006,775

Table 1: Number of words in each partition of the generic text corpus per language. The ES+EU corpus is a concatenation of the data from EU and ES.

5.2 Mintzai-ST corpus

As it was previously mentioned, the final models of both BRNN-based and BERT-based systems were trained and evaluated with the mintzai-ST corpus, which incorporates both textual and acoustic features. This corpus is composed by a collection of manual transcriptions of proceedings of the Basque

Parliament from 2011 to 2018, which comprises content in Basque and Spanish. The original train, development and test partitions of the corpus were maintained in both languages as they are described in its related paper. Table 2 presents the amount of words in each partition.

	EU	ES	ES+EU
train	1,137,727	4,468,041	5,605,768
dev	13,672	25,139	38,811
test	40,644	75,470	116,114

Table 2: Number of words in each partition of the corpus mintzai-ST per language. The ES+EU corpus is a concatenation of the data from EU and ES.

The original mintzai-ST corpus was processed in order to represent the information needed to train models. This information is related to each word and consists of four elements: the lowercase word itself, the punctuation label, a float value representing the silence duration, and the capitalisation label. The word and the silence value act as inputs, while the other two are the outcomes the system should learn to predict. An example of a completely annotated sentence from the corpus mintzai-ST can be found in Figure 4.

```
primer          O             0.00   FIRST_CAP
punto           O             0.00   O
del             O             0.00   O
orden           O             0.00   O
del             O             0.00   O
día             COLON         0.00   O
pregunta        OPEN_QUOTE    0.00   FIRST_CAP
formulada       O             0.00   O
por             O             0.00   O
don             O             0.00   O
andoni          O             0.00   FIRST_CAP
ortuzar         O             0.00   FIRST_CAP
arruabarrena    COMMA         0.36   FIRST_CAP
```

Figure 4: An example of the training corpus. Each word has its corresponding punctuation, acoustic and capitalisation labels respectively.

5.3 Punctuation labels

The punctuation labels represent the punctuation marks that should go attached to each word. The current punctuation labels inventory is the following: `COLON` (`:`), `COMMA` (`,`), `DASH` (`W-`), `ELLIPSIS` (`…`), `EXCLAMATION` (`!`), `OPEN_DASH` (`-W`), `OPEN_EXCL` (`¡`), `OPEN_QUES` (`¿`), `OPEN_QUOTE` (“), `PERIOD` (`.`), `QUESTION` (`?`), `QUOTE` (”) and `SEMICOLON` (`;`). There is also an `O` label to indicate words that bear no punctuation. A single word can have more than one punctuation label attached to it.

The punctuation labels were derived from the different Unicode code-points present in the original transcriptions of the dataset, e.g. the code-points `U+00BB` (») and `U+201D` (”) are both labelled as `QUOTE`.

Table 3 shows the distribution in percentages of the punctuation labels for Basque (EU), Spanish (ES) and Spanish+Basque (ES+EU). The distributions shows a label unbalance. This is to be expected since some punctuation marks, such as periods or commas, are much more frequent than the others.

label	EU	ES	ES+EU
COMMA	54.62%	59.89%	58.35%
PERIOD	36.25%	28.02%	30.41%
QUESTION	1.87%	1.72%	1.76%
COLON	1.88%	1.49%	1.61%
DASH	1.21%	1.54%	1.44%
OPEN_QUES	0.02%	1.68%	1.20%
SEMICOLON	1.08%	1.20%	1.16%
OPEN_QUOTE	1.15%	1.09%	1.11%
QUOTE	1.02%	1.05%	1.04%
ELLIPSIS	0.69%	1.08%	0.97%
EXCLAMATION	0.17%	0.62%	0.49%
OPEN_EXCL	0.01%	0.61%	0.43%
OPEN_DASH	0.03%	0.00%	0.01%

Table 3: Percentage of appearance of each punctuation label in each language in the corpus mintzai-ST.

5.4 Capitalisation labels

The capitalisation labels consist in two different options: `FIRST_CAP` if the first letter of the word is a capital letter, and `ALL_CAPS` if the whole word is written in capital letters. Similarly to the punctuation, the label `O` indicates that the word is not capitalised. For words that do not fall into any of these categories (e.g. EiTB), these same criteria are similarly applied: if the first letter is cased it would carry the label `FIRST_CAP`, and `O` otherwise.

6 Evaluation and discussion

In this section, we present the results obtained by each neural architecture proposed in this work on the test partition of the mintzai-ST corpus. In the case of our BERT-based system, it is composed of several elements and parameters that can be enabled

or disabled to build the models. During our experiments, we constructed several models with different combinations in order to evaluate the impact of these elements:

- Using or ignoring silences duration.
- Modelling silences as continuous or discrete values.
- Weighting silence embeddings uniformly or using a Gaussian distribution (only applies to experiments with discrete silences).
- Using or skipping the additional Transformer layer (Custom BERT).

6.1 Punctuation results

Table 4 shows the micro-averaged F_1 score for each experiment with AutoPunct, and the comparison with Punctuator.

S	D	f	B	EU	ES	ES+EU
×	–	–	×	76.2	78.2	75.4
×	–	–	✓	**76.9**	78.5	75.6
✓	×	–	×	76.8	78.4	75.9
✓	×	–	✓	76.6	77.7	75.2
✓	✓	G	×	76.8	78.4	76.0
✓	✓	G	✓	76.7	**78.9**	**76.4**
✓	✓	U	×	76.8	78.5	76.3
✓	✓	U	✓	**76.9**	**78.9**	76.3
BRNN				74.6	68.7	72.9

Table 4: Micro-averaged F_1 scores for AutoPunct and Punctuator in each language. S: Using information of silences. D: Using discrete silences. f: Gaussian (G) or uniform (U) distribution for contiguous buckets. B: Adding a custom BERT. The results achieved by the BRNN-based system are also displayed.

As it can be observed in Table 4, AutoPunct obtains better aggregated scores for the three language scenarios. The scores show a moderate improvement towards the lower part of the table, where discrete silences are used in combination with the intermediate custom BERT.

In Table 5, the evaluation results for each individual punctuation label are displayed. As it can observed, the BRNN-based system obtains a higher F_1 score than AutoPunct in the label `PERIOD` for both Basque and Spanish+Basque. Nevertheless, the rest of punctuation marks are better modelled by the BERT-based architectures. Moreover, the score for some of these labels using the BRNN-based system is 0.00%, such as in `QUOTE` or `OPEN_QUOTE` in Basque, in contrast with AutoPunct reaching F_1 scores higher than 50%. This can be due to the fact that in the first corpus used to train the BRNN-based system (Section 5.1) does not contain these labels, as well as to their low appearance rate in the corpus of mintzai-ST.

As it can be appreciated, for the Basque language, the absence of predicted `OPEN_QUES` and `OPEN_EXCL` indicates a correct behaviour, since such punctuation marks are not used in this language. For Spanish, in contrast, `OPEN_QUES` reaches a 64.8%. In the case of the labels with lower F_1 scores, it seems that the number of occurrences in this dataset are not enough for a proper training and evaluation.

Finally, in the case of the Spanish+Basque model there is a small performance loss, but it can be considered a reasonable trade-off for performing punctuation and capitalisation in two languages simultaneously using a single module. Furthermore, it is not uncommon in Basque to interleave Spanish words or sentences spontaneously in casual conversations, so using a model that deals with both languages at the same time can be advantageous on these scenarios.

6.2 Capitalisation results

The capitalisation results are presented just for AutoPunct, since the BRNN-based model does not perform this task. The micro-averaged F_1 scores for automatic capitalisation evaluation are shown in Table 6, following the same experiment breakdown.

As it can be seen in Table 6, the obtained micro-averaged F_1 values are very high in all the cases and for every language scenario. The variations in the architecture do not show a high impact in the final scores.

6.3 Inference speed

In addition to evaluating the quality of the systems in the automatic punctuation and capitalisation tasks, parameters like speed and efficiency are also desirable properties. To assess if the proposed system would be suitable for a scenario requiring real-time predictions, we measured the rate of words per second at inference time. This measure has been carried on during the evaluation, using the test set for each language. The eval-

Basque (EU)

S	D	f	B	COM	PER	QUES	COL	DASH	O_QS	SCOL	O_QUO	QUO	ELL	EXC	O_EX
×	–	–	×	74.6	84.8	58.8	**53.8**	13.6	0.0	**17.9**	59.3	47.6	0.0	0.0	0.0
×	–	–	✓	**75.7**	85.5	60.4	50.9	**20.3**	0.0	15.7	55.3	**54.5**	8.3	0.0	0.0
✓	×	–	×	75.3	85.8	61.0	51.2	0.0	0.0	0.0	54.4	40.0	0.0	0.0	0.0
✓	×	–	✓	75.5	85.0	63.5	47.1	3.6	0.0	0.0	50.7	43.9	0.0	0.0	0.0
✓	✓	G	×	75.4	85.6	**65.5**	49.8	14.5	0.0	10.9	58.1	51.2	0.0	0.0	0.0
✓	✓	G	✓	75.2	85.9	60.8	48.5	14.5	0.0	12.9	55.1	51.5	**11.6**	0.0	0.0
✓	✓	U	×	75.2	85.5	64.5	50.6	5.4	0.0	6.0	56.4	46.7	0.0	0.0	0.0
✓	✓	U	✓	75.5	86.1	62.0	50.0	13.6	0.0	10.3	**59.9**	47.3	0.0	0.0	0.0
BRNN				70.1	**87.2**	52.2	23.4	0.0	0.0	12.4	0.0	0.0	0.0	0.0	0.0

Spanish (ES)

S	D	f	B	COM	PER	QUES	COL	DASH	O_QS	SCOL	O_QUO	QUO	ELL	EXC	O_EX
×	–	–	×	79.8	85.7	58.5	49.8	17.6	64.2	30.6	48.9	26.9	5.4	5.6	10.7
×	–	–	✓	80.1	86.1	56.5	48.1	**21.7**	63.8	33.3	50.4	32.7	**16.3**	**13.3**	**24.1**
✓	×	–	×	79.8	86.8	56.3	**51.6**	11.9	61.3	23.8	48.4	25.3	0.0	2.9	0.0
✓	×	–	✓	79.7	85.1	56.6	46.1	12.3	63.3	25.5	50.0	20.2	0.0	0.0	5.9
✓	✓	G	×	80.1	86.5	54.1	45.0	17.9	62.0	27.1	47.7	31.1	0.0	2.9	5.7
✓	✓	G	✓	**80.2**	87.0	**59.7**	46.8	20.4	**64.8**	31.3	51.9	33.0	0.0	2.9	16.0
✓	✓	U	×	**80.2**	86.3	57.0	48.4	16.9	63.4	30.5	48.2	29.4	1.8	2.9	5.7
✓	✓	U	✓	80.0	**87.3**	59.0	48.2	21.4	64.0	**33.9**	**52.2**	**34.2**	0.0	5.7	16.2
BRNN				65.4	85.6	46.0	11.9	1.8	23.0	1.7	13.0	0.0	5.2	11.9	0.0

Spanish+Basque (ES+EU)

S	D	f	B	COM	PER	QUES	COL	DASH	O_QS	SCOL	O_QUO	QUO	ELL	EXC	O_EX
×	–	–	×	75.6	83.7	**56.9**	48.6	14.7	57.5	19.3	53.6	41.7	13.2	14.9	16.7
×	–	–	✓	75.9	83.7	55.7	48.6	21.1	59.9	20.8	55.2	**44.8**	**23.1**	**24.6**	**33.0**
✓	×	–	×	75.9	84.6	54.1	**51.7**	10.9	57.0	18.3	54.7	37.7	12.7	11.4	10.8
✓	×	–	✓	75.3	83.7	**56.9**	50.2	21.6	**60.3**	20.3	54.1	43.8	20.1	15.7	21.3
✓	✓	G	×	76.1	84.5	55.4	48.6	7.6	57.1	19.8	52.4	36.4	10.4	9.3	13.9
✓	✓	G	✓	**76.5**	85.4	56.2	51.5	18.1	58.3	21.3	55.3	40.8	18.5	19.0	24.0
✓	✓	U	×	76.3	85.0	55.8	51.3	16.4	57.9	22.1	54.4	40.7	17.4	13.3	24.1
✓	✓	U	✓	76.2	85.6	56.2	50.2	**23.7**	58.8	**24.8**	**55.5**	41.7	19.0	21.4	32.7
BRNN				69.7	**87.4**	49.9	16.1	2.0	3.3	9.1	6.2	0.0	13.6	5.2	0.0

Table 5: Class-wise F_1 scores for AutoPunct and the BRNN-based system in each language. Labels with a F_1 score of 0.0 in the three language scenarios were omitted. S: Using information of silences. D: Using discrete silences. f: Gaussian (G) or uniform (U) distribution for contiguous buckets. B: Adding a custom BERT.

S	D	f	B	EU	ES	ES+EU
×	–	–	×	91.7	92.1	90.6
×	–	–	✓	91.6	92.1	90.9
✓	×	–	×	91.9	92.2	91.4
✓	×	–	✓	91.6	91.7	90.7
✓	✓	G	×	91.7	92.1	91.3
✓	✓	G	✓	91.9	92.3	91.5
✓	✓	U	×	91.6	92.1	91.4
✓	✓	U	✓	**92.2**	**92.4**	**91.7**

Table 6: Capitalisation micro-averaged F_1 scores. S: Using information of silences. D: Using discrete silences. f: Gaussian (G) or uniform (U) distribution for contiguous buckets. B: Adding a custom BERT.

uations have been run using a Nvidia GeForce RTX 2080ti GPU[1]. Again, we compare AutoPunct with the BRNN-based system. These results are shown in Table 7.

From the results of the table 7, it can be observed that the computation speed is fast enough to enable a real-time processing. The trend in the numbers show that the use of discrete silences requires more time, but this is not surprising due to the extra amount of computation to select and average the silence embeddings. The same reasoning applies to the intermediate Custom BERT. Compared to the Punctuator baseline, all the architectural variations of the proposed system are faster, in special taking into account that

[1]These numbers should be taken as approximations, since different hardware or different workloads may lead to slightly different results.

S	D	f	B	EU	ES	ES+EU
×	–	–	×	6221	**7379**	**7572**
×	–	–	✓	6454	7038	6576
✓	×	–	×	**6802**	7112	7537
✓	×	–	✓	5749	7186	6749
✓	✓	G	×	3097	3315	3128
✓	✓	G	✓	3071	3322	3068
✓	✓	U	×	2877	3124	3206
✓	✓	U	✓	3017	3174	3169
BRNN				1729	2126	2524

Table 7: Processing speed in words per second. S: Using information of silences. D: Using discrete silences. f: Gaussian (G) or uniform (U) distribution for contiguous buckets. B: Adding a custom BERT.

AutoPunct is also adding capitalisation to the output in the same process.

7 Conclusions

In this work we present AutoPunct, an automatic punctuation and capitalisation system based on BERT that also makes use of the silence duration between words. The system was trained for 13 different punctuation labels and two types of capitalisation. It works as a multilabel classifier, so it can predict several punctuation marks attached to a single word. The model is language agnostic and only depends on training data, it can be even trained on several languages at the same time. Due to its inference speed, ranging from 3000 to 7000 words per second depending on the chosen architectural variation, it can be used in real-time scenarios. The system was tested in Spanish and Basque, both individually and simultaneously, using the mintzai-ST dataset. We carried on experiments with several architectural variations to assess their impact in the final result. We also compared the proposed system with another well known automatic punctuation system, Punctuator from (Tilk and Alumäe, 2016), showing not only better results but also faster inference times. As future work, we would like to test additional architectural variations and hyper-parameters, and also train and evaluate on datasets of more varied styles and application domains.

Ackowledgments

This work was supported by the Department of Economic Development and Competitiveness of the Basque Government under projects GAMES (ZL-2020/00074) and DeepText (KK-2020-00088).

References

Agerri, R., I. S. Vicente, J. A. Campos, A. Barrena, X. Saralegi, A. Soroa, and E. Agirre. 2020. Give your text representation models some love: the case for basque. *arXiv preprint arXiv:2004.00033*.

Alam, T., A. Khan, and F. Alam. 2020. Punctuation restoration using transformer models for resource-rich and-poor languages. pages 132–142.

Cañete, J., G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Che, X., C. Wang, H. Yang, and C. Meinel. 2016. Punctuation prediction for unsegmented transcript based on word vector. pages 654–658.

Christensen, H., Y. Gotoh, and S. Renals. 2001. Punctuation annotation using statistical prosody models.

Courtland, M., A. Faulkner, and G. McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. pages 272–279.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, June.

Etchegoyhen, T., H. Arzelus, H. Gete Ugarte, A. Alvarez, A. González-Docasal, and E. Benites Fernandez. 2021. mintzai-ST: Corpus and Baselines for Basque-Spanish Speech Translation. In *Proc. IberSPEECH 2021*, pages 190–194.

Federico, M., M. Cettolo, L. Bentivogli, P. Michael, and S. Sebastian. 2012. Overview of the iwslt 2012 evaluation campaign. In *IWSLT-International Workshop on Spoken Language Translation*, pages 12–33.

Jones, D. A., F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. A. Reynolds, and M. Zissman. 2003. Measuring the readability of automatic speech-to-text transcripts. In *Eighth European Conference on Speech Communication and Technology*.

Kim, J.-H. and P. C. Woodland. 2003. A combined punctuation generation and speech recognition system and its performance enhancement using prosody. *Speech Communication*, 41(4):563–577.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and F. Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019)*, pages 1–18.

Nassif, A. B., I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. 2019. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165.

Otegi, A., A. Agirre, J. A. Campos, A. Soroa, and E. Agirre. 2020. Conversational question answering in low resource scenarios: A dataset and case study for basque. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 436–442.

Peitz, S., M. Freitag, A. Mauser, and H. Ney. 2011. Modeling punctuation prediction as machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2011*.

Schuster, M. and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Sunkara, M., S. Ronanki, K. Dixit, S. Bodapati, and K. Kirchhoff. 2020. Robust prediction of punctuation and truecasing for medical asr. pages 53–62.

Tilk, O. and T. Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. pages 3047–3051.

Yi, J. and J. Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. pages 7270–7274.

Yi, J., J. Tao, Y. Bai, Z. Tian, and C. Fan. 2020a. Adversarial transfer learning for punctuation restoration. *arXiv preprint arXiv:2004.00248*.

Yi, J., J. Tao, Z. Tian, Y. Bai, and C. Fan. 2020b. Focal loss for punctuation prediction. *Proc. Interspeech 2020*, pages 721–725.