Participation of HULAT-UC3M in SEPP-NLG 2021 shared task

Jose Manuel Masiello-Ruiz
Computer Science Dept.
Univ. Carlos III de Madrid
jmasiell@eco.uc3m.es

Jose Luis Lopez Cuadrado
Computer Science Dept.
Univ. Carlos III de Madrid
jllopez@inf.uc3m.es

Paloma Martinez
Computer Science Dept.
Univ. Carlos III de Madrid
pmf@inf.uc3m.es

Abstract

This paper introduces the HULAT-UC3M system developed to participate in the SEPP-NLG 2021 shared task. The systems is based on the Punctuator framework, a bidirectional recurrent neural network model with attention mechanism for automatic punctuation trained on the Europarl dataset provided by organizers. The best results obtained in Subtask 1 are F1 score of 84%, 79%, 36% and 83% for EN, IT, DE and FR languages on development dataset, respectively. Concerning Subtask 2, F1 score are 63%, 57%, 69% and 64% for EN, IT, DE and FR languages on development dataset, respectively.

1 Introduction

Automatic punctuation is a relevant task when it comes to processing text obtained from transcription systems. When transcription is made using Automatic Speech Recognition (ASR) systems, the punctuation marks are not always available or, when they are available, they must be reviewed. Detecting the end of phrases or the punctuation mark to be included in a specific position of the text improves the readability and preserves its meaning. When the transcriptions are large raw text documents, the process is not affordable by people. This paper presents the HULAT-UC3M system developed to participate in the SEPP-NLG 2021 shared task. The aim of the system is, on the one hand, to detect the full stop marks in the text by training the punctuator Matusov et al. (2006) framework with the Europarl dataset provided for the shared task. On the other hand, in the context of subtask 2, the trained framework will be tested on the detection of full punctuation marks.

The remainder of this paper is organized as follows: Section 2 summarizes the relevant related work for the proposal, Section 3 presents the proposed system, Section 4 describes and discusses the results obtained, and Section 5 presents the conclusions and the future work.

2 Background

Automatic generation of punctuation marks from the output of an ASR system has many applications such as enhance dictation systems avoiding that the speaker verbalizes special keywords to add punctuation marks (comma, colon, semicolon, question mark, etc.) to the text or to enhance readability of captions in content broadcasting. Some previous related research concerning automatic punctuation of texts is summarized in this section. System described in Chen (1999) is based on a method that combines acoustic and lexical evidence. The hypothesis is although acoustic pauses do not match one to one with linguistic segmentation, the combination of acoustic and lexical information allows a good prediction of punctuation marks. This system used the IBM speech recognizer trained on 1,800 speakers and with speaker adaptation and a N-gram model built using 250 million words. Using 4 scenarios that consider different types of pauses, the best performance considering punctuation mark at correct place and of correct type is 57%. The test dataset used was a letter with 333 word with 31 punctuation marks read by three speakers.

Work described in Matusov et al. (2006) was a similar approach in the context of machine translation, considering that it is easier to predict segment boundaries taking into account prosodic features and pauses of different length than predicting if a punctuating marks should be inserted than a word position. Using a HMM model the system achieved a F-measure of 70% (results are worse with spontaneous speech). For Portuguese language, Batista et al. (2008) used maximum entropy n-grams with features such as lexical features (POS tags, words) and acoustic features (time, speaker change among others); testing on broadcast news the system got 83% of precision and 61% of recall for full stop recovering and worse performance for comma recovery (45% of precision and 16% of recall).

More recently,Öktem et al. (2017) proposed using recurrent neural networks trained on TED talks to predict punctuation marks (with similar features of previous works- words, pause, frequency and intensity values of words, etc). Best performance of this system is F score of 65.7% for all comma, period and question marks. Finally, Sunkara et al. (2020) introduces pretrained BERT language models fine-tuned to the medical domain data to improve automatic punctuation and truecasing prediction. This approach was tested using two medical datasets (dictation and conversational) and the best F score was 93% for full stop trained on wiki and medical dictation data and 82% for full stop trained on wiki and medical conversation data.

By reviewing the previous related works, approaches that combine lexical and acoustic features integrated in current deep learning architectures could provide better results to cope with the problems of ASR errors and out of vocabulary words.

3 System description

To respond to the proposed tasks we have used Punctuator, an implementation of a bidirectional recurrent neural network with attention mechanism introduced by Ottokar Tilk and Tanel Alumäe Tilk and Alumäe (2016) (https://github.com/ottokart/punctuator2).

Punctuator has been adapted to take into account the set of proposed punctuation marks: ”: -,?. 0”. The adaptation of the data format to the one expected by Puntuactor has been carried out with a previous pre-process.

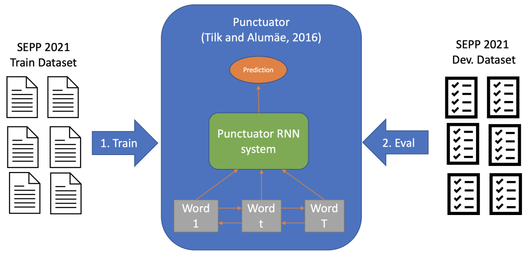

Figure 1: Proposed system for the tasks

Eight models have been trained, one for each task and language. All models have been configured with a 256 hidden layer size and a 0.02 learning rate. The data set sepp_nlg_2021_train_dev_data_v5.zip have been used as training and dev data set.

4 Results

4.1 Experiment setup

We have used a google cloud server with the following configuration:

- 4 CPU virtuals, 15G memory.
- 1 GPU NVIDIA Tesla K80.
- Ubuntu pro 16.04.
- Python 3.8.
- CUDA 10.2.
- CNN 7.6.5.
- Theano 1.0.5.

For training we have used the data sets sepp_nlg_2021_train_dev_data_v5.zip and for evaluating we have used the data sets sepp_nlg_2021_test_data_unlabeled_v5 where there are two data sets: test and surprise test.

4.2 Data pre-processing

For task 1, both data sets (dev and train column 1), have been processed in the same way. All the training .tsv files have been merged into a single language.train.txt file where each sentence is a line and the mark ”.” has been replaced by ”.PERIOD”. Likewise, a language.dev.txt file has been generated from the dev .tsv files.

There is a test data set that is a copy of language.test.txt file.

For task 2 the marks (column 2 of the data sets) have been mapped as shown in Table 1

Mark	Mapped
,	,COMMA
.	.PERIOD
?	?QUESTIONMARK
:	:COLON
-	-DASH

Table 1: Task 1 training characteristics.

In the same way as task 1 a language.train.txt, language.dev.txt and language.test.txt files have been generated from .tsv trainning, test and dev files.

For the evaluation the pre-processing is the same but we have used the sepp_nlg_2021_test_data_unlabeled_v5 data sets.

4.3 Subtask 1 Results

For each language we have trained a model with the following characteristics in Table 2:

lang.	hidden layers	learning rate	train file	dev file
en	256	0.02	en.train.txt	en.dev.txt
it	256	0.02	it.train.txt	it.dev.txt
de	256	0.02	de.train.txt	de.dev.txt
fr	256	0.02	fr.train.txt	fr.dev.txt

Table 2: Task 1 training characteristics

We have tested each models with its test.txt (or dev.txt) file and the results are shown in Tables 3, 4, 5, 6:

	en			
	prec.	recall	f1-score	support
0	0.99	0.99	0.99	7422156
1	0.86	0.81	0.84	321333
accur.			0.99	7743489
macro avg	0.93	0.90	0.91	7743489
weighted avg	0.99	0.99	0.99	7743489

Table 3: English.Task 1 results.

	it			
	prec.	recall	f1-score	support
0	0.99	1.00	0.99	6904100
1	0.86	0.73	0.79	290089
accuracy			0.98	7194189
macro avg	0.92	0.86	0.89	7194189
weighted avg	0.98	0.98	0.98	7194189

Table 4: Italian.Task 1 results.

For each of the unlabeled files of the data set (selecting column 1 of the .tsv files), a prediction file .tsv has been generated using its corresponding model according to language.

	de			
	prec.	recall	f1-score	support
0	0.99	0.85	0.92	6067240
1	0.23	0.90	0.36	291443
accuracy			0.85	6358683
macro avg	0.61	0.87	0.64	6358683
weighted avg	0.96	0.85	0.89	6358683

Table 5: Deutsche.Task 1 results.

	fr			
	prec.	recall	f1-score	support
0	0.99	1.00	0.99	8449263
1	0.87	0.79	0.83	332330
accuracy			0.99	8781593
macro avg	0.93	0.89	0.91	8781593
weighted avg	0.99	0.99	0.99	8781593

Table 6: French.Task 1 results.

4.4 Subtask 2 Results

For each language we have trained a model with the following characteristics Table 7:

lang.	hidden layers	learning rate	train file	dev file
en	256	0.02	en.train.txt	en.dev.txt
it	256	0.02	it.train.txt	it.dev.txt
de	256	0.02	de.train.txt	de.dev.txt
fr	256	0.02	fr.train.txt	fr.dev.txt

Table 7: Task 2 training characteristics.

We have tested each models with its test.txt (or dev.txt) file and the results are shown in Tables 8, 9, 10, 11. The figures 2, 3, 4, 5, shown the confusion matrix for each language.

For each of the unlabeled files of the data set (selecting column 2 of the .tsv files), a prediction file .tsv has been generated using its corresponding model according to language.

4.5 Discussion

Regarding Subtask 1, learning rates are the same in the four languages. The evaluation is based on the

	en			
	prec.	**recall**	**f1-score**	**support**
,	0.73	0.70	0.72	401095
-	0.53	0.07	0.12	18335
.	0.85	0.86	0.86	319751
0	0.98	0.99	0.99	6985003
:	0.64	0.23	0.34	9815
?	0.80	0.72	0.76	9490
accuracy			0.96	7743489
macro avg	0.76	0.60	0.63	7743489
weighted avg	0.96	0.96	0.96	7743489

Table 8: English. Task 2 results.

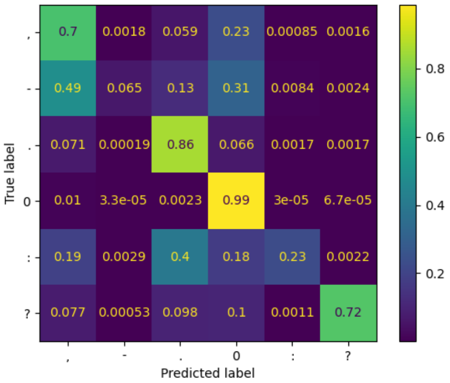

Figure 2: Task 2. English. Confusion matrix.

	it			
	prec.	**recall**	**f1-score**	**support**
,	0.73	0.63	0.67	385867
-	0.44	0.05	0.09	13044
.	0.84	0.83	0.83	290088
0	0.98	0.99	0.98	6480166
:	0.58	0.27	0.37	14658
?	0.73	0.37	0.49	10366
accuracy			0.96	7194189
macro avg	0.72	0.52	0.57	7194189
weighted avg	0.95	0.96	0.96	7194189

Table 9: Italian. Task 2 results.

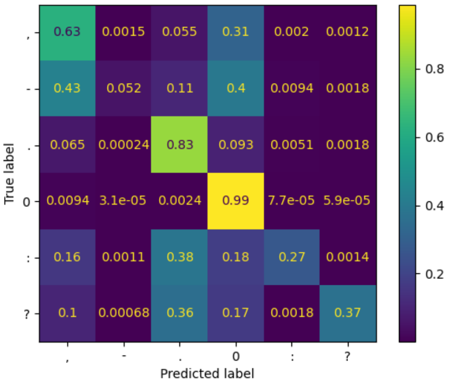

Figure 3: Task 2. Italian. Confusion matrix.

	de			
	prec.	**recall**	**f1-score**	**support**
,	0.90	0.89	0.90	489257
-	0.50	0.09	0.15	17412
.	0.92	0.92	0.92	287680
0	0.99	1.00	0.99	5544080
:	0.63	0.36	0.46	11148
?	0.83	0.65	0.73	9106
accuracy			0.98	6358683
macro avg	0.79	0.65	0.69	6358683
weighted avg	0.98	0.98	0.98	6358683

Table 10: Deutsche. Task 2 results.

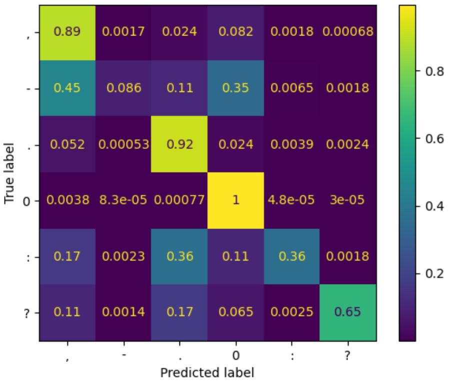

Figure 4: Task 2. Deutsche. Confusion matrix.

value 1 (full stop) in each language. The F1-score in English is 0.84, but the framework presents similar F-scores in French (0,85). In Italian, the result is slightly worst, 0.79, and the worst result is in German with 0.36. When comparing the results of Subtask 2, with the same learning rates, the value of the F-score for the full stop is 0,86 in English and French, 0,83 in Italian and 0.92 in German.

	fr			
	prec.	**recall**	**f1-score**	**support**
,	0.75	0.71	0.73	445852
-	0.49	0.07	0.11	18321
.	0.86	0.85	0.86	328795
0	0.98	0.99	0.99	7964631
:	0.60	0.32	0.42	12482
?	0.82	0.63	0.71	11512
accuracy			0.97	8781593
macro avg	0.75	0.59	0.64	8781593
weighted avg	0.97	0.97	0.97	8781593

Table 11: French. Task 2 results.

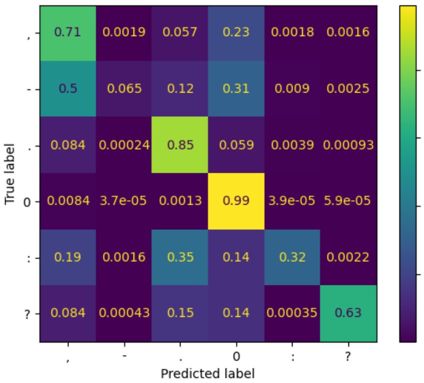

Figure 5: Task 2. French. Confusion matrix.

The difference for German between subtask 1 and subtask 2 is remarkable. Regarding the rest of the punctuation marks in subtask 2, the worst results in all languages are obtained in the dash mark followed by the colon (:). Remarkably, the proposed framework obtain, for subtask 2 in the four languages, the best overall measures (accuracy, macro average and weighted average) for German.

5 Conclusions and Future Work

The approach presented in this paper is an exploratory participation in the SEPP-NLG 2021 task. We are interested in automatic segmentation and punctuation for Spanish spontaneous speech. We plan to use BETO, the Spanish version of BERT Vaswani et al. (2017) and mBERT models by integrating different types of word embeddings to face the out-of-vocabulary problem.

Acknowledgments

This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

References

Fernando Batista, Diamantino Caseiro, Nuno Mamede, and Isabel Trancoso. 2008. Recovering capitalization and punctuation marks for automatic speech recognition: Case study for Portuguese broadcast news. *Speech Communication*, 50:847–862.

Julian C. Chen. 1999. Speech recognition with automatic punctuation. *Sixth European Conference on Speech Communication and Technology*, (January):6–9.

Evgeny Matusov, Arne Mauser, and Hermann Ney. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *International Workshop on Spoken Language Translation (IWSLT) 2006*.

Alp Öktem, Mireia Farrús, and Leo Wanner. 2017. Attentional parallel RNNs for generating punctuation in transcribed speech. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10583 LNAI:131–142.

Monica Sunkara, Srikanth Ronanki, Kalpit Dixit, Sravan Bodapati, and Katrin Kirchhoff. 2020. Robust prediction of punctuation and truecasing for medical asr. In *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, pages 53–62.

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 08-12-Sept(September):3047–3051.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.