INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Joint Learning of Correlated Sequence Labeling Tasks Using Bidirectional Recurrent Neural Networks

Vardaan Pahuja[1], Anirban Laha[1*], Shachar Mirkin[2], Vikas Raykar[1], Lili Kotlerman[2], Guy Lev[2]*

[1]IBM Research - Bangalore, India
[2]IBM Research - Haifa, Israel

vapahuja@in.ibm.com, anirlaha@in.ibm.com, shacharm@il.ibm.com, viraykar@in.ibm.com, lili.kotlerman@il.ibm.com, guylev@il.ibm.com

Abstract

The stream of words produced by Automatic Speech Recognition (ASR) systems is typically devoid of punctuations and formatting. Most natural language processing applications expect segmented and well-formatted texts as input, which is not available in ASR output. This paper proposes a novel technique of jointly modeling multiple correlated tasks such as punctuation and capitalization using bidirectional recurrent neural networks, which leads to improved performance for each of these tasks. This method could be extended for joint modeling of any other correlated sequence labeling tasks.

1. Introduction

Sequence labeling involves the assignment of a categorical label to each element of a sequence of tokens. Some common examples include punctuation prediction for automatic speech recognition (ASR) transcripts, capitalization recovery (i.e. restoring the case of the lowercased words, a.k.a. *truecasing*), part-of-speech tagging (POS), and named entity recognition (NER). In this work we address the task of *multiple sequence labeling*, where the goal is to assign *multiple categorical labels* to each element of the sequence, such as predicting both the punctuation and capitalization for a given ASR speech transcript. We specifically address the scenario in which the multiple sequence labeling tasks are correlated. Consider the following two examples:

1. *. . . and it hasn't been refined enough yet**. It** needs to be worked on until it can speak fluently*
2. *This young doctor**, Tom Ferguson,** was the medical editor of the Whole Earth Catalog.*

The first example shows the occurrence of capitalization preceded by a period. In the second example, the two commas surround capitalized proper nouns. Such co-occurrences illustrate the fact that punctuation and capitalization are two correlated tasks that could benefit from each other. We refer to these kinds of sequence labeling tasks as *correlated multiple sequence labeling* and propose a novel approach using a bidirectional recurrent neural network (BiRNN) [1], that is trained jointly for prediction across such tasks.

Speeches are often transcribed by ASR systems that convert the audio signals into a stream of words. Apart from often having a high word error rate, this stream is also devoid of the standard textual structure present in written texts. These structural aspects [2] include punctuation, capitalization, and numeric data formatting, such as for digits, dates, and phone numbers. Recovering the structure from raw word transcripts is essential for two main reasons. First, the structure enhances the readability and understanding of the transcripts [3, 4]. Second, its recovery enables subsequent text processing and makes it more accurate. Many works have shown the impact of the structure recovery for tasks such as summarization [5, 6], part-of-speech (POS) tagging [7, 8], named entity recognition (NER) [7], machine translation [9, 10] and information extraction [11], among others.

We attempt to recover two aspects of structure, punctuation and capitalization, by casting it as correlated multiple sequence labeling problem. Earlier work [12] proposed the idea of training multiple sequence labeling tasks together, and showed a slight improvement for POS and NER when combined with task-specific feature engineering. However, they assumed the availability of sentence segmentation and capitalization as inputs. The solution we propose does not assume any feature engineering and is suitable for speech transcripts, which do not come with punctuation or capitalization.

Multiple papers [2, 4, 13, 14, 15, 16, 17, 18] showed the usefulness of pause duration and prosodic features for punctuation prediction as compared to using textual features alone. In this work, our goal is to boost the accuracy of punctuation prediction without taking into account additional inputs such as prosodic features; we accomplish this by training the capitalization task jointly with the punctuation task. To the best of our knowledge, this is the first RNN (BiRNN)-based framework for joint training of correlated sequence labeling tasks. Moreover, this framework is general enough to be applicable for jointly training other correlated sequence labeling tasks such as POS tagging and NER.

In a nutshell, our contributions are the following:

- An RNN (BiRNN)-based joint learning framework for multiple correlated sequence labeling tasks, with no feature engineering.
- Improvement in punctuation prediction on speech transcripts by jointly training it with capitalization, without using any prosodic features. A similar improvement is also observed in capitalization.
- State-of-the-art performance on benchmark punctuation prediction dataset.

2. Correlated Multiple Sequence Labeling

Punctuation and capitalization are considered highly important for the structure recovery of ASR transcripts. There are various effective approaches to insert punctuation and specifically sentence boundaries into raw speech transcripts [19, 20]. In this work, we consider both punctuation and capitalization together,

*Equal contribution by the first two authors.

http://dx.doi.org/10.21437/Interspeech.2017-1247

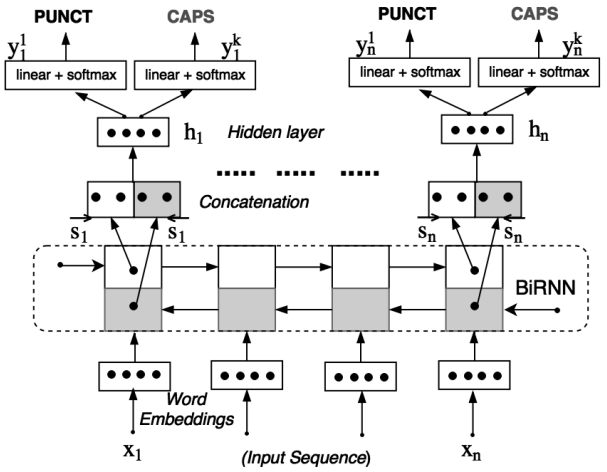

Figure 1: *Framework for Correlated Sequence labeling Tasks*

treating it as a *correlated multiple sequence labeling* problem, as defined below:

> *Given a sequence of words $W = (w_1, w_2, w_3, ..., w_n)$ from a vocabulary V, the objective is to predict K labels $\{l_i^1, l_i^2, ..., l_i^K\}$ corresponding to each word w_i, one label for each of the K sequence labeling tasks. This will produce K correlated output sequences of the form $O^k = (l_1^k, l_2^k, ..., l_n^k)$. Here, labels for different tasks come from different label spaces, as in $l_i^k \in L^k$.*

Following the above definition, $K = 1$ trivially implies a *single sequence labeling problem*. In our setting, $K = 2$ when we consider the punctuation and capitalization tasks together. Typically, three punctuation marks have received the most attention in existing literature due to their high frequency of occurrence: periods, commas, and question marks (Q-MARK below). Thus, L^1 ={COMMA, PERIOD, Q-MARK, NO-PUNCT}, where there is a high class imbalance tilted towards the NO-PUNCT class. In our model formulation, the label l_i^1 corresponds to the punctuation occurring before the word w_i. As for capitalization, the label l_i^2 determines the surface form of word w_i, which can be any of the following: all-lowercase (e.g., 'hello'), all-uppercase (e.g., 'NASA'), mixed-case (e.g., 'McGill'), sentence-case (only the first letter capitalized, e.g., 'London') and single-letter-word-case (e.g., 'I').

Given a sequence of n input vectors $\mathbf{x}_1, ..., \mathbf{x}_n$ and an initial state vector $\mathbf{s}_0$, an RNN generates a sequence of n state vectors $\mathbf{s}_1, ..., \mathbf{s}_n$ alongside a sequence of n output vectors $\mathbf{y}_1, ..., \mathbf{y}_n$; that is, $RNN(\mathbf{s}_0, \mathbf{x}_1, ..., \mathbf{x}_n) = \mathbf{s}_1, ..., \mathbf{s}_n, \mathbf{y}_1, ..., \mathbf{y}_n$. The input vectors $\mathbf{x}_i$ are the latent embeddings (word2vec [21]) of the words w_i in the sequence, and $\mathbf{s}_i$ represents the state of the RNN after observing the inputs $\mathbf{x}_1, ..., \mathbf{x}_i$. The output vector $\mathbf{y}_i$ is a function of the corresponding state vector $\mathbf{s}_i$ and is then used for prediction of output labels for the correlated tasks. An RNN is defined by the following update equations: $\mathbf{s}_i = R(\mathbf{x}_i, \mathbf{s}_{i-1})$ and $\mathbf{y}_i = O(\mathbf{s}_i)$. Different instantiations of R and O will result in different network structures (Simple RNN, LSTM [22], GRU [23], etc.).

A bidirectional RNN consists of two parallel RNNs: one running forward and another running backward. These capture the context in both directions (since the words to the right have significant influence on a word label in addition to the words to its left). Essentially, the same sequence of input vectors $\mathbf{x}_1, ..., \mathbf{x}_n$ is fed to both RNNs to produce the sequence of state vectors $\overrightarrow{\mathbf{s}_1}, ..., \overrightarrow{\mathbf{s}_n}$ from the forward RNN and $\overleftarrow{\mathbf{s}_1}, ..., \overleftarrow{\mathbf{s}_n}$ from the backward RNN. Here we extend the bidirectional RNN to model multiple correlated sequence labeling tasks together. For the k-th task being considered, the output sequence, denoted by $\mathbf{y}_1^k, ..., \mathbf{y}_n^k$, can be derived from the sequence of state vectors $\mathbf{s}_1, ..., \mathbf{s}_n$, where $\mathbf{s}_i = [\overrightarrow{\mathbf{s}_i}, \overleftarrow{\mathbf{s}_i}]$, through transformations, as defined below:

$$\mathbf{h}_i = f(\mathbf{s}_i) = \phi(\mathbf{W}\mathbf{s}_i + \mathbf{b}) \quad (1)$$

$$\mathbf{m}_i^k = g^k(\mathbf{h}_i) = \mathbf{W}^k\mathbf{h}_i + \mathbf{b}^k \quad (2)$$

$$\mathbf{y}_i^k = softmax(\mathbf{m}_i^k) \quad (3)$$

In the above formulation, the concatenated state vector $\mathbf{s}_i$ is transformed linearly and passed through the function $\phi \in \{sigmoid, tanh, relu, linear\}$ to produce a hidden layer vector $\mathbf{h}_i$. To produce outputs for the different correlated tasks in question, the vector $\mathbf{h}_i$ is then passed through different branches of $linear + softmax$, one branch for each of the tasks. That is, for the k-th task, the output $\mathbf{y}_i^k$, is produced from the k-th branch, which leads to the prediction of label l_i^k. The set of trainable parameters are $\{\mathbf{W}, \mathbf{b}, \{\mathbf{W}^k, \mathbf{b}^k\}_{k=1}^K\}$ in addition to the parameters defining the forward and backward RNNs. Figure 1 illustrates our model formulation.

Joint Training Loss Function: The network formulated above is defined for multiple correlated tasks (say K tasks) and is capable of producing K sequences of outputs of the form $\mathbf{y}_1^k, ..., \mathbf{y}_n^k$. While predicting the outputs for the different tasks, all the trainable parameters required until the computation of $\mathbf{h}_i$ are shared across all tasks and are trained jointly based on the loss function defined over the outputs of all K tasks. We compute the loss $\mathcal{L}^k$ for every task using the standard cross-entropy loss function. Then, based on predefined weights q_k (over tasks), a weighted average of task-specific losses is taken to produce the final loss $\mathcal{L}$ to be optimized:

$$\mathcal{L} = \sum_{k=1}^{K} q_k \mathcal{L}^k \quad (4)$$

This accumulated loss helps the network predict well across all tasks. If the tasks are correlated (as in our case), then each task should help the other tasks through the joint learning of shared parameters. These shared parameters help produce *correlated representations* $\mathbf{h}_i$, which can be used to generate predictions for all tasks.

3. Experiments

To corroborate the hypothesis that our jointly trained model helps improve performance over the individual tasks, we experimented on two different datasets, as described below. All our models[1] are evaluated based on precision (P), recall (R) and F_1 score, for each punctuation class, and overall for all classes, as well as with Slot Error Rate (SER)[2] [24].

3.1. Datasets

Intelligence Squared: This dataset was obtained from the Intelligence Squared (IQ2 henceforth) debating television show, whose transcripts are publicly available.[3] We used 45 debates, each containing talks by four speakers, from which we created a train-validation-test split in a ratio of 60:10:30.[4]

[1]The code will be available at `https://goo.gl/3UGd4p`

[2]SER is the ratio of the total number of slot errors (substitutions, deletions, and insertions) in the predicted set of labels, to the total number of slots in the gold set of labels.

[3]`http://www.intelligencesquaredus.org/`

[4]Evaluated on reference transcripts only as ASR is not available.

IWSLT TED Talks: We used the English transcripts of the English-to-French machine translation task in IWSLT 2012[5] as our training data with the same train-validation splits as suggested in [25]. We report our test results on two datasets[6]: the first, used by [26] (henceforth referred to as test-set-1), is the development data of the IWSLT 2011 ASR and SLT tasks[7]; the second test set (henceforth referred to as test-set-2) consists of test-dataset-2 of the IWSLT 2011 ASR and SLT tasks, as used by [25] and T-BRNN [27].

3.2. Experimental Setup

Data Preprocessing: Each training sequence consists of a random number of tokens (40 to 70 in our experiments), with the constraint that it must begin with a new sentence. The unfinished sentence forms the beginning of next training sequence. This scheme of generating training sequences prevents the model from always learning to predict a period or a question mark at the end of every sequence. For the validation and test datasets, we used a single consolidated sequence comprising all the sentences, to simulate a real ASR stream. This is not done for the training dataset to avoid memory issues with extremely long sequences. To evaluate our model on ASR transcripts, we mapped the punctuations and capitalization from the reference transcripts to the ASR transcripts, based on Levenshtein alignment, as discussed in [13]. Since the mapping process is sensitive to ASR word errors, we adopted the approach in [26], and restricted the evaluation to only those punctuations for which the left and right context words have been recognized correctly by the ASR. Similarly, we restricted capitalization evaluation to the words matching in the reference. For punctuation, we used the standard four classes as mentioned in Section 2, whereas for capitalization, sentence-case and mixed-case were merged as the latter occurs very rarely.

Network Training and Tuning: We trained our model architecture using standard backpropagation in TensorFlow [28]. In our experiments, we trained two kinds of models: joint model (or Corr-BiRNN), which was trained jointly on punctuation and capitalization tasks, and task-specific models (or Single-BiRNN), that were trained separately for each of the two tasks. We carried out extensive hyper-parameter tuning for both the joint model and the separate task-specific models, for the IQ2 and TED datasets. The tuned hyper-parameters included: the number of layers and the number of hidden units per layer in the BiRNN, RNN dropout rate, RNN output dropout rate, type of RNN (Simple RNN, LSTM or GRU), the number of units in the outer hidden layer, hidden layer activation function, task-specific loss weights, and batch size. The best hyper-parameter setting for the joint model as well as the task-specific models was selected based on SER performance on the validation set for the task at hand. We then evaluated the selected settings on the reference transcripts of the corresponding test sets and on ASR test set (available for TED only) for the respective tasks. Note that the best hyper-parameter setting for a punctuation task-specific model may not be the same as that of a capitalization task-specific model. In other words, Single-BiRNN may have different settings selected based on the task at hand. Similarly, for Corr-BiRNN, different settings give best validation SER performance on punctuation and capitalization tasks.

[5] https://wit3.fbk.eu/mt.php?release=2012-03

[6] Both Reference transcripts and ASR

[7] http://iwslt2011.org/doku.php?id=06_evaluation

4. Results and Discussion

The test evaluations are reported in Tables 1-3. For all tables, each row contains test evaluation metrics for the hyper-parameter setting that was selected based on validation SER performance of the task being considered. Table 4 shows example outputs of our models, on ASR compared to gold labels, created by mapping from reference transcripts. These are shown separately for each of the two tasks for better illustration.

For the IQ2 dataset (see Table 1), the joint training results in improved performance on both of the tasks, as compared to models trained for each of the individual tasks. This is consistent based on both overall F_1-score as well as SER metrics.

For the TED capitalization task (Table 3), the Corr-BiRNN model outperforms the Single-BiRNN model performance in terms of F_1 score for all test sets across both reference and ASR transcripts (this includes test-set-1 and test-set-2, though results are shown only for test-set-2 in the interest of space). However, improvement is not seen in UPPERCASE performance; this may be explained by the fact that this label does not correlate with any punctuation.

Regarding the TED punctuation task (Table 2), the Corr-BiRNN model outperforms Single-BiRNN (F_1 score) for the punctuation task for test-set-1 (Ref.), test-set-1 (ASR) and test-set-2 (Ref.); that is, in three out of four cases. For test-set-2 (ASR), the Single-BiRNN model is marginally better than the Corr-BiRNN model. This is possibly due to the reason that reference transcripts were used for validation (due to unavailability of ASR for validation), because of which the best hyper-parameter setting might have been missed for both models.

While comparing to the existing baselines on the TED punctuation task (see Table 2), our Corr-BiRNN model fares significantly better on all fronts (especially Q-MARK with 22.9% gain in F_1 score), compared to the existing baseline [26] for test-set-1 (Ref.). In fact, its performance on test-set-1 (ASR) is better than the baseline for test-set-1 (Ref.). It also outperforms the T-BRNN [27] baseline in terms of COMMA and PERIOD for test-set-2 (Ref.), which are the more frequent punctuations, in addition to overall, measured in F_1 score. For test-set-2 (ASR) though, we do not see improvement, again possibly because the validation dataset is based on reference transcripts.

Despite having a much simpler model, in many cases we were able to beat the baseline performance of T-BRNN [27], a more complex attention-based BiRNN model. This substantiates our claim that joint learning helps learning better representations than task-specific training for a particular task. Our simpler model has the added value of learning and predicting much faster than T-BRNN. In addition, our predictions are generated in one shot over the whole consolidated test sequence and does not need to follow a window based prediction as in T-BRNN.

5. Related Work

Simple approaches for single sequence labeling include unigram [8] and n-gram language models [29]. These models see limited fixed context around a word which may not be sufficient for prediction and they also face data sparsity issues as n increases. There are also classical approaches like Hidden Markov Models (HMM), maximum-entropy models (Max-Ent) and conditional random fields (CRF), all of which try to model a hidden state sequence corresponding to the observed word sequence as in [19, 30, 26, 31, 15, 17, 2]. However, these models are more difficult to train and construction of hand-crafted features is non-trivial. Models built using deep neural networks

	Task	Model	*Class Labels*												
			COMMA			**PERIOD**			**Q-MARK**			**OVERALL**			
Ref.	*Punctuation*		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	SER
	Single-BiRNN		43.7	**54.9**	**48.7**	**73.9**	19.3	30.6	**52.3**	23.7	32.6	48.0	39.0	43.0	77.6
	Corr-BiRNN		**57.9**	34.3	43.1	62.0	**53.3**	**57.3**	45.8	**25.7**	**32.9**	**59.7**	**42.0**	**49.3**	**68.9**
			UPPERCASE			**SENTENCE-CASE**			**SINGLE-CASE**			**OVERALL**			
Ref.	*Capitalization*		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	SER
	Single-BiRNN		**96.5**	**63.2**	**76.4**	**87.0**	55.7	67.9	**99.9**	**98.2**	**99.0**	**89.6**	61.5	72.9	45.3
	Corr-BiRNN		95.1	**63.2**	76.0	80.9	**65.3**	**72.3**	99.7	98.0	98.9	84.2	**69.5**	**76.2**	**43.0**

Table 1: *Intelligence Squared (IQ2) results*

	Model	**COMMA**			**PERIOD**			**Q-MARK**			**OVERALL**			
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	SER
Ref.	Ueffing et al.[26]	(45.0)	(**47.0**)	(46.0)	(54.0)	(72.0)	(62.0)	(53.0)	(33.0)	(41.0)	(47.8)	(54.8)	(51.0)	-
	T-BRNN [27]	64.4	45.2	53.1	72.3	71.5	71.9	67.5	58.7	62.9	68.9	58.1	63.1	51.3
	T-BRNN-pre [27]	**65.5**	47.1	54.8	73.3	**72.5**	72.9	**70.7**	**63.0**	**66.7**	**70.0**	59.7	64.4	**49.7**
	Single-BiRNN	62.2	47.7	54.0	74.6	72.1	**73.4**	67.5	52.9	59.3	69.2	59.8	64.2	51.1
		(**58.1**)	(41.4)	(48.4)	(72.2)	(72.0)	(72.1)	(76.9)	(**59.5**)	(**67.1**)	(**66.1**)	(55.5)	(60.3)	(**58.1**)
	Corr-BiRNN	60.9	**52.4**	**56.4**	**75.3**	70.8	73.0	**70.7**	56.9	63.0	68.6	**61.6**	**64.9**	50.8
		(55.6)	(44.5)	(**49.4**)	(**72.5**)	(**72.2**)	(**72.4**)	(74.6)	(56.0)	(63.9)	(64.5)	(**57.1**)	(**60.6**)	(59.2)
ASR	Ueffing et al. [26]	-	-	-	-	-	-	-	-	-	-	-	-	-
	T-BRNN [27]	**60.0**	45.1	51.5	69.7	69.2	69.4	61.5	45.7	52.5	65.5	57.0	60.9	57.8
	T-BRNN-pre [27]	59.6	42.9	49.9	**70.7**	**72.0**	**71.4**	60.7	48.6	54.0	**66.0**	57.3	**61.4**	**57.0**
	Single-BiRNN	55.9	48.7	52.0	63.1	70.9	66.8	**66.7**	**50.0**	**57.1**	60.1	59.6	59.8	64.1
		(**45.7**)	(35.6)	(40.0)	(60.2)	(**67.4**)	(**63.6**)	(**56.4**)	(53.7)	(55.0)	(**53.7**)	(50.1)	(51.8)	(76.0)
	Corr-BiRNN	53.5	**52.5**	**53.0**	63.7	68.7	66.2	**66.7**	**50.0**	**57.1**	59.0	**60.3**	59.7	65.4
		(44.9)	(**40.6**)	(**42.6**)	(**61.4**)	(64.8)	(63.1)	(56.1)	(**56.1**)	(**56.1**)	(53.2)	(**51.7**)	(**52.4**)	(**75.7**)

Table 2: *TED punctuation results over test-set-2 and test-set-1 (in parentheses).*

	Model	**UPPERCASE**			**SENTENCE-CASE**			**SINGLE-CASE**			**OVERALL**			
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	SER
Ref.	Single-BiRNN	**94.1**	**64.0**	**76.2**	**84.4**	68.2	75.4	**100.0**	98.9	99.4	**88.8**	75.3	81.5	33.8
	Corr-BiRNN	93.7	60.0	73.2	82.6	**71.9**	**76.9**	99.4	**99.7**	**99.6**	87.2	**78.2**	**82.4**	**33.0**
ASR	Single-BiRNN	87.5	87.5	87.5	**80.4**	58.6	67.8	**100.0**	99.1	99.5	**86.7**	69.2	76.9	**41.3**
	Corr-BiRNN	**87.5**	**87.5**	**87.5**	76.3	**62.2**	**68.6**	99.4	**100.0**	**99.7**	83.3	**72.1**	**77.3**	42.3

Table 3: *TED capitalization results on test-set-2.*

	Punctuation	**Capitalization**
Gold	I ended up hiking up Mount Kilimanjaro **,** the highest mountain in Africa .	I wish you luck . **May** none of your non cancer cells become endangered species .
Single-BiRNN	i ended up hiking up mount kilimanjaro . the highest mountain in africa .	I wish you luck **may** none of your non cancer cells become endangered species
Corr-BiRNN	I ended up hiking up Mount Kilimanjaro **,** the highest mountain in Africa .	I wish you luck . **May** none of your non cancer cells become endangered species .

Table 4: *Examples of joint vs. task-specific model predictions on the TED ASR dataset.*

(DNNs) [12, 25] usually consist of a context window around the word being considered, which is fed to a multi-layer perceptron that extracts different abstractions of features relevant to the sequence-labeling task. More recent approaches [4, 30, 27] considered RNNs, especially LSTMs and reported good results. Compared to the fixed window-based approaches, LSTMs can work on the full sequence of words and dynamically adapt their internal representations. These papers have shown deep learning based solutions outperform the classical approaches.

Multiple sequence labeling tasks and their inter-dependence has been studied in great detail [12]. However, for tasks like POS tagging, NER and chunking, they assumed the availability of punctuation and capitalization, which is not true for ASR transcripts. More recently, joint prediction of punctuation and capitalization for transcribed speech has been attempted in [29], albeit using n-gram language models. In [32], a joint label space for punctuation and capitalization tasks is created, in order to predict labels for both tasks. This is, however, not scalable since label space can possibly explode with the introduction of more labels for each task. A few other works related to joint sequence labeling include joint parsing and punctuation prediction [33] using a CRF-based model, and disfluency detection alongside other NLP tasks like punctuation prediction [17] and dependency parsing [34], using classical solutions. In our work, we explore the joint learning of correlated sequence labeling tasks like punctuation and capitalization using a deep-learning based approach without any feature engineering being involved.

6. Conclusion

In this paper, we have shown the utility of models jointly trained on two correlated tasks, punctuation and capitalization, to learn better representations for each of them. Our simple jointly-trained BiRNN model, trained only on lexical features, outperforms several complex models, which demonstrates its robustness and generalization ability. Future work will involve the joint training of a variety of other correlated NLP tasks like POS tagging and NER.

7. References

[1] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[2] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1526–1540, 2006.

[3] M. Shugrina, "Formatting time-aligned asr transcripts for readability," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Association for Computational Linguistics, 2010, pp. 198–206.

[4] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] Y. Liu and S. Xie, "Impact of automatic sentence segmentation on meeting summarization," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5009–5012.

[6] J. Mrozinski, E. W. Whittaker, P. Chatain, and S. Furui, "Automatic sentence segmentation of speech for automatic summarization," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[7] D. Hillard, Z. Huang, H. Ji, R. Grishman, D. Hakkani-Tur, M. Harper, M. Ostendorf, and W. Wang, "Impact of automatic comma prediction on pos/name tagging of speech," in *2006 IEEE Spoken Language Technology Workshop*. IEEE, 2006, pp. 58–61.

[8] L. V. Lita, A. Ittycheriah, S. Roukos, and N. Kambhatla, "Truecasing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 152–159.

[9] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence segmentation and punctuation recovery for spoken language translation," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5105–5108.

[10] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation." in *IWSLT*, 2006, pp. 158–165.

[11] B. Favre, R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tür, and M. Ostendorf, "Punctuating speech for information extraction," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5013–5016.

[12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[13] J. Kolár and L. Lamel, "Development and evaluation of automatic punctuation for french and english speech-to-text." in *INTERSPEECH*, 2012, pp. 1376–1379.

[14] J.-H. Kim and P. C. Woodland, "The use of prosody in a combined system for punctuation generation and speech recognition." in *INTERSPEECH*, 2001, pp. 2757–2760.

[15] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech." in *INTERSPEECH*, 2002.

[16] T. Levy, V. Silber-Varod, and A. Moyal, "The effect of pitch, intensity and pause duration in punctuation detection," in *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*. IEEE, 2012, pp. 1–4.

[17] D. Baron, E. Shriberg, and A. Stolcke, "Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues," *Channels*, vol. 20, no. 61, p. 41, 2002.

[18] V. Eidelman, Z. Huang, and M. Harper, "Lessons learned in part-of-speech tagging of conversational speech," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. Association for Computational Linguistics, 2010, pp. 821–831.

[19] X. Wang, H. T. Ng, and K. C. Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction." in *INTERSPEECH*, 2012, pp. 1384–1387.

[20] C. Xu, L. Xie, G. Huang, X. Xiao, E. Chng, and H. Li, "A deep neural network approach for sentence boundary detection in broadcast news." in *INTERSPEECH*, 2014, pp. 2887–2891.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 3111–3119.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, 2014, pp. 1724–1734.

[24] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel *et al.*, "Performance measures for information extraction," in *Proceedings of DARPA broadcast news workshop*, 1999, pp. 249–252.

[25] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), May 2016.

[26] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *INTERSPEECH*, F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, Eds. ISCA, 2013, pp. 3097–3101.

[27] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," *Interspeech 2016*, pp. 3047–3051, 2016.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*, 2016.

[29] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 4741–4744.

[30] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, August 2016, pp. 1064–1074.

[31] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 177–186.

[32] T. Baldwin and M. P. A. K. Joseph, "Restoring punctuation and casing in english text," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2009, pp. 547–556.

[33] D. Zhang, S. Wu, N. Yang, and M. Li, "Punctuation prediction with transition-based parsing." in *ACL (1)*, 2013, pp. 752–760.

[34] M. Honnibal and M. Johnson, "Joint incremental disfluency detection and dependency parsing," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 131–142, 2014.