Punctuation Restoration for Ukrainian Broadcast Speech Recognition System based on Bidirectional Recurrent Neural Network and Word Embeddings

Mykola Sazhok[a], Anna Poltieva[b], Valentyna Robeiko[a,b], Ruslan Seliukh[a] and Dmytro Fedoryn[a]

[a] *International Research/Training Center for Information Technology and Systems, Kyiv, Ukraine*
[b] *Taras Shevchenko National University, Kyiv, Ukraine*

Abstract

The output of a speech-to-text conversion system is a sequence of words accomplished, optionally, with the speaker turn information. Lack of punctuation complicates reading for humans and degrades the performance of many downstream machine processing tasks. We investigated various punctuation restoration models based on Deep Learning for Ukrainian. The chosen tool uses Bidirectional Recurrent Neural Network to generate probabilities for hypothetically placed punctuation marks. The selected text input is based on publicly available text data. Experimentally was chosen an efficient way of text processing applied to the dataset. Significant improvement for punctuation generation accuracy was achieved with word embedding application. The broadcast transcribing system with supported punctuation restoration is presented.

Keywords [1]

Punctuation restoration, punctuation generation, automatic speech recognition, Ukrainian language, recurrent neural network, word embeddings.

1. Introduction

To become consumable products, automatic speech recognition (ASR) systems have taken a long road of evolving.

Advances in Ukrainian ASR, particularly, for broadcast domain (such as tackling a language changing problem as well as speaker diarization and numeric text processing), allowed for thousands and thousands of broadcast hours to become searchable and the preparation of desired episode transcripts now can be done more efficiently [1], [2]. At last, punctuation, used in the right place, would significantly improve the readability of transcripts and reduce the final transcript preparation time. Furthermore, many downstream machine processing tasks without punctuation marks are quite problematic.

Without sentence segmentation, it is impossible to determine the boundaries of the statement, which is the bearer of the communicative function [3]. Although, the punctuation absence is not a significant obstacle for tasks like sentiment analysis or named entity recognition [4], for other tasks the presence of punctuation is essential. First of all, punctuation significantly increases the human perception of the text. Punctuation presence is important for official documents, such as clinical dictations (which is a standard procedure in Western countries) [5] or transcripts of court hearings. Finally, punctuation significantly increases the effectiveness of many types of NLP tasks such as semantic parsing, question-answer systems, machine translation etc [6].

1COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: sazhok@gmail.com (Mykola Sazhok); poltyeva.anna@gmail.com (Anna Poltieva); valya.robeiko@gmail.com (Valentyna Robeiko); vxml12@gmail.com (Ruslan Seliukh); dmytro.fedoryn@gmail.com (Dmytro Fedoryn)
ORCID: 0000-0003-1169-6851 (Mykola Sazhok); 0000-0001-8537-0390 (Anna Poltieva); 0000-0003-2266-7650 (Valentyna Robeiko); 0000-0003-2230-8746 (Ruslan Seliukh); 0000-0002-4924-225X (Dmytro Fedoryn)

Known approaches are based on solely text analysis or on both lexical and prosody modeling and have been evaluated mostly for English. Since the corpus with prosody is unavailable for Ukrainian, we focused on text-only approaches. Furthermore, unlike English, Ukrainian is a highly inflective language with relatively free word order, which complicates the punctuation restoration process. To overcome this we considered vocabulary reduction and word embedding direct usage as well as various text processing techniques.

Wide range of domains should be covered for broadcast. The fact that broadcast contains plenty of dialogs should be taken into consideration as well. Therefore, we omit focusing on a specific domain and require modeling of realistic spontaneous dialog.

Though we understood that covering most punctuation marks are not feasible we consider the minimal set of punctuation characters hoping its more or less accurate restoration may help to make automatic transcripts more usable.

The remainder of the paper is structured as follows. In Section 2 we describe the structure of punctuation marks in Ukrainian, then we justify the tool choice in Section 3 describe the related approach Section 4. Experimental research is covered in Section 5, which includes data acquisition and preparation, model tuning as well as experimental evaluation and applied system is presented in Section 6. Conclusion and future outlook are in Section 7.

2. Punctuation in Ukrainian

Punctuation marks are conditionally accepted graphic signs, the main purpose of which is to divide the written form of language to express and understand the content better [3].

The following main punctuation marks are distinguished in Ukrainian:

- end of sentence (EOS) marks: period, ellipsis (can also be inside the sentence), exclamation mark, question mark;
- characters within a sentence: comma, semicolon, colon, dash, hyphen, brackets, parentheses, quotation marks.

The term "punctuation" in linguistics has several meanings [7]. It denotes:

1. system of graphic punctuation marks;
2. historically established system of rules for the use of punctuation in writing;
3. a section of linguistics that studies the laws of punctuation and codified rules for the use of punctuation.

In our experimental study, we will use the term "punctuation" in the first meaning, as well as its synonymous term "punctuation marks". However, to define the rules of punctuation mark usage, in this section of our work, we will refer to the term "punctuation" meaning "a section of linguistics".

The following principles of punctuation are distinguished in the Ukrainian language [3]:

- **Syntactic** principle indicates the syntactic structure of the sentence and its units. For example, in the sentence *If I have time, I will call you* a the main clause is separated by a comma, although in spoken language it may not be expressed with physical break, especially, for the fast speech rate.
- **Morphological** principle indicates the morphological nature of the members of the sentence. For example, in the sentence "*Сакура – окраса нашого саду*" (*Sakura is the adornment of our garden*) a dash in Ukrainian variant is placed between the subject and the predicate expressed by the noun, and in the sentence "*Вона окраса нашого саду*" (*It is the adornment of our garden*) there is no dash because here the subject is expressed not by a noun but by a pronoun. This principle is not relevant for English, but is proper for Ukrainian.
- **Semantic** principal states that punctuation depends on the content of the sentence. So, in the sentence I saw a man, eating lobster a comma is separating a clause whilst in I saw a man-eating lobster man-eating is one word and the meaning of a sentence is completely different.
- **Intonation** principle is closely connected with the semantic one, because with the help of intonation we either form the meaning of the sentence (*Good. Good? Good!!!*), or specify the meaning of the statement, e.g., *подув вітер, зашелестіло листя* with token by token translation: *wind blew, leaves rustled* (a sequence of events), *подув вітер – зашелестіло листя*: *wind blew – leaves rustled* (a cause-and-effect development).

The main principles of Ukrainian punctuation are considered to be structural (morphological and syntactic) and semantic. The intonational principle, which is "programmed" by the content of the sentence, is auxiliary. These factors greatly complicate the process of automatic punctuation recognition, because when processing the natural language, including the speech to text transformation, it is easier to determine the movement of the fundamental frequency and the structure of intonation, while it is much more difficult to determine the morphological and syntactic structure of words and sentences, and semantic properties of text are usually not marked at all.

3. Related Work

For Ukrainian language, there is no open-source system that would automatically place punctuation marks, however, there are proofreading tools in text processing software and services [8], [9] that may suggest some punctuation correction in certain cases. All of them only partially recognize punctuation errors. Obviously, such systems are not applicable for punctuation restoration in ASR output. To create such a system, firstly, we must build a classifier that will automatically determine the punctuation.

Several open-source tools allow for training the model by text with possible inclusion some prosodic features [10]–[14]. Only models trained on same IWSLT speech dataset [15] are comparable. The corpus consists of 1046 talks by 884 English speakers, uttering a total amount of 156034 sentences that is about 1000 hours of speech. The corresponding transcripts, as well as audio and video files, are available on TED's website; they were created by volunteers and include punctuation and paragraph breaks [16]. More than 9% better overall F_1-score was demonstrated by *PunkProse* tool [14] comparing to the *Punctuator*2 results [11] in terms of absolute differences. *PunkProse* allows for the integration of any desired lexical, syntactic or prosodic feature [14].

It is worth to note that certain proprietary captioning systems, available for a few languages, may support punctuation and description of such systems is valuable for further insights [17],[18].

As of our knowledge, a large enough Ukrainian speech corpus with verified annotated punctuation is currently unavailable. Therefore, in this work, no prosodic cues, even pauses in speech signal, could be taken into account. Hence, we have chosen *Punctuator*2 [11] that is a punctuation restoration tool based on bidirectional recurrent neural network (BRNN) that can be trained on punctuated text optionally accomplished with duration of breaks between words.

4. Method

The method used in *Punctuator*2, BRNN [11], enables it to make use of unfixed length contexts before and after the current position in text. Gated recurrent units (GRU) are used in the recurrent layers, having similar benefits as long short-term memory (LSTM) units, while being simpler [19] The incorporated attention mechanism [20] increases the capacity of finding relevant parts of the context for punctuation decisions. To fuse together the model state at current input word and the output from the attention mechanism a late fusion approach [21] is used. This allowed the attention model output to directly interact with the recurrent layer state while not interfering with its memory. The neural network general structure is illustrated in Figure 1.

GRU consists of bidirectional layers for words, W_e. The hidden states of the GRU layers at time step t are:

$$\vec{h}(t) = GRU\left(x_t W_e, \vec{h}(t-1)\right),$$

where x_t is a word index at time step t. This vector is concatenated with a similarly expressed backward direction vector to form the bidirectional context vector $h(t)$ to be passed over as input to another unidirectional layer:

$$s(t) = GRU\left(h(t), s(t-1)\right).$$

At time step t the model outputs probabilities for punctuation x_t to be placed between the previous word x_{t-1} and current input word $h(t)$. As there is no punctuation before the first word x_1, the model predicts punctuation only for words $x_2, \ldots, x_T$ where x_T is a special utterance termination token.

The attention mechanism is useful for the neural network to identify positions in a sequence where important information is concentrated. For words, it helps to focus on positions of words and word combinations that signal the introduction of a punctuation mark.

The output GRU layer uses a late-fusion approach, which lets the context gradient carry on easily by preventing it passing through many activation functions.

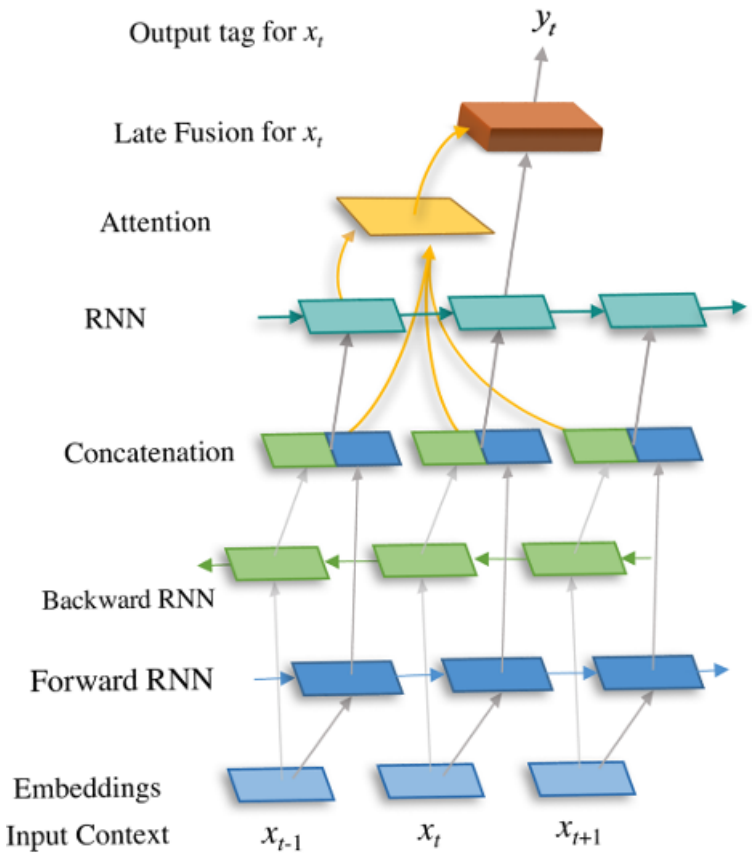

Figure 1: Neural network design for punctuation restoration. The diagram shows an input context for the word x_t and the stack of layers that result in the tag $h(t)$ representing the punctuation decision for x_t [5].

5. Experimental research

In this section, we present the experimental research that consists of dataset preparation, various text preprocessing as well as word embedding incorporation followed by the evaluation the trained models. The metrics for evaluation are Precision, Recall and F_1-score for each punctuation mark and overall assessment. Also the latter is accomplished with slot error rate (SER), since F_1-score has been shown to have certain undesirable properties such as deweighted deletion and insertion error by a factor of two [22].

5.1. Input Preparation

A text corpus, consisting (in the final preprocessing) of 1,217,443 tokens (including punctuation marks), 1,024,639 tokens (excluding punctuation marks), 124,592 of which are unique, has been collected mainly from the subset of Brown-Uk corpus that is well prepared and verified in accordance

with the Brown principles and verified [23]. Since Brown-Uk corpus lacks for dialogs, we added texts containing plenty of conversations taken from interviews and literature samples. The prepared corpus was divided into training, development and test sets in the ratio of approximately 80/10/10 (just over 10 MB for training and almost 1.5 MB for development and for test set).

The entire text was converted to uppercase (this was done in order not to give the model case hints, because if the text contains a word with a capital letter, then with a very high probability we can say that this is the beginning of the sentence, and before this word there should be a punctuation mark of the end of the sentence), removed tags, links in the form of URLs, as well as links to literature in square brackets. The numbering of document items and sections has been removed, ellipsis have been replaced with a period, and other minor garbage deletions were done. All punctuation marks have been converted to the format specified by the toolkit requirements, namely:

,COMMA, .PERIOD, ?QUESTIONMARK, !EXCLAMATIONMARK, :COLON, -DASH.

Quotation marks and parentheses were tokenized separately, in order to be able to analyze their influence on basic punctuation restoration. In fact, only three first punctuation marks were analyses in most cases, due to few examples of others in the training set.

5.2. Models

In order to better understand how certain changes in text preprocessing affect the performance of the model, a number of models were trained. The trained models are summarized in Table 1. The initial modification to *baseline* model consisted in number processing. Since numbers usually do not carry additional information [24], they were replaced with a special <NUM> token. Further, all quotes and parentheses were removed etc.

The pretrained word embeddings for Ukrainian are available at the website of the community of applied linguists lang-uk [25]. 300-d vectors are generated for 3 different corpora in the form of Word2Vec, LexVec and GloVe embeddings. Another source is fastText website with available embeddings for 157 languages [26].

Below we briefly characterize modifications made for each trained model:

1. *baseline* – basic text cleaning
2. *no digits* – digits are replaced with <NUM> tag
3. *no quotes* – quotations and parentheses are removed
4. *new preprocessing* – unification of similar symbols like quotes and dashes
5. *no numbers* – all remaining tokens related to numbers replaced with <NUM>
6. *ubercorpus lexvec* – word embeddings based on LexVec model and Ubertext corpus
7. *fastText embeddings* - fastText 300-d word embeddings

The number of out-of-vocabulary words (UNK tokens) is in the range of 7-10% in the train set and 12-15% in development and test sets. Vocabularies of pretrained embeddings do not cover the entire lexicon the dataset. Therefore, models that integrate pretrained embeddings have smaller vocabulary size and larger out-of-vocabulary rate as follows by Table 1.

Proceeding from the prepared input, *Punctuator*2 models were trained to automatically detect punctuation in an unsegmented text.

5.3. Evaluation

The outcomes of our experiments in restoring periods, commas and question marks are presented in Table 2 in terms of precision (P), recall (R), and F_1 scores. The overall scores are accomplished with slot error rate (SER). As it follows from Table 2, *baseline* model failed restoring question marks, whereas all other models were able to detect it, although, still most of question marks are skipped as indicated by their low recalls.

Table 1

Summary of model configurations in numbers

No	Model name	Vocabulary	Number of	Out-of-vocabulary rate per sample (%)

		size	parameters	Train	Development	Test
1	baseline	50905	15068679	7.43	12.18	12.29
2	no digits	50625	14996999	7.39	12.11	12.23
3	no quotes	50540	14975239	7.47	12.23	12.35
4	new preprocessing	50552	14978568	7.32	11.99	12.11
5	no numbers	50573	14983687	7.39	12.12	12.23
6	ubercorpus lexvec	47535	16365383	10.59	14.77	14.83
7	fastText embeddings	48465	16644383	9.32	13.66	13.74

A significant improvement is achieved with the proposed word embeddings is expressed in 6.8% of better F_1 and reduced 7.5% of SER. *Ubercorpus lexvec* model with top metrics in most categories, including overall SER, notably concedes to *ubercorpus lexvec* model in full stop sensitivity and F_1-score.

Table 2
Punctuation restoration results for reference transcripts

Model name	Comma			Period			Question			Overall			
	P	R	F_1	P	R	F_1	P	R	F	P	R	F_1	SER
baseline	70.8	41.4	52.3	40.6	33.6	36.8	0.0	0.0	-	59.4	36.1	45.0	73.7
no digits	71.1	42.5	53.2	42.5	31.0	35.8	42.9	1.2	2.0	61.1	36.3	45.5	72.9
no quotes	58.8	54.8	56.8	41.9	**35.0**	38.1	37.5	3.1	5.8	54.0	**45.5**	49.4	75.6
new preprocessing	52.6	**59.4**	55.8	46.6	23.4	31.2	**64.0**	3.1	6.0	51.5	39.5	44.7	79.2
no numbers	60.6	52.5	56.3	48.6	3.5	6.6	48.6	3.5	6.6	52.6	40.8	45.9	76.8
ubercorpus lexvec	68.5	57.4	**62.5**	**59.9**	27.0	37.3	53.8	**17.8**	**26.8**	**66.1**	42.6	**51.8**	**66.2**
fastText embeddings	**71.3**	51.5	59.8	51.4	34.9	**41.6**	47.1	9.8	16.2	64.6	41.0	50.2	68.5

Thus, in our work the best model used the pretrained embeddings taken from the website of the community of applied linguists lang-uk [25]. We used embeddings from fiction (filtered to fit our vocabulary), since the tokens from there better match the tokens from our vocabulary (approximately 47,500 out of 50,500). The model showed the best results: on the set test, the F_1-score is 51.8% and SER is 66.2%.

6. Application

In this section we describe the progress in development of the system that move towards an efficient transcribing of the Ukrainian broadcast media to meet many needs of individuals and companies that consume and analyze the extracted content with integrated punctuation restoration. Figure 2 illustrates the architecture of speech-to-text conversion for the broadcast media transcribing. A recognition component, the actual Recognizer, receives a speech signal extracted form media data at the input and, at the output, referring to the Data and Knowledge base (D and KB) produces a Recognition Response.

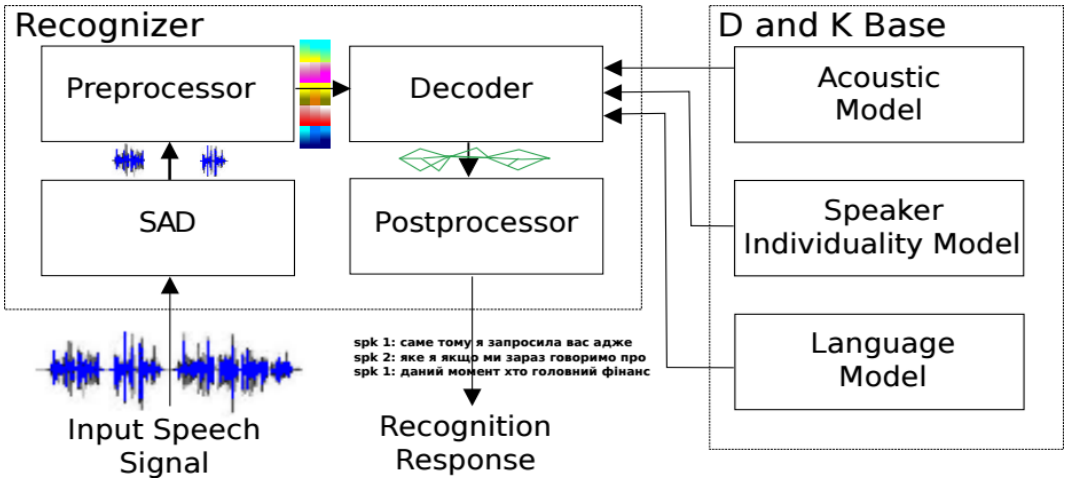

Figure 2: Diagram of speech-to-text conversion architecture.

Input Speech Signal passing through the Speech Activity Detector (SAD) is segmented by speech presence or absence [27]. For each segment where speech activity is detected, Preprocessor converts the waveform into the feature space based on mel-frequency cepstral coefficients supplemented with the i-vector in accordance to the speaker adaptation technique (SAT) [28]. The latter allows also for completing the speaker diarization procedure that estimates probabilities of speech transition from one person to another [29].

Decoder estimates the similarity criteria value for all valid model signal hypotheses given the input signal, which is memorized in the Dynamic Programming graph referred as lattice. To speed-up the decoding process, on decoding stage, the lexical context is limited to bigrams and most frequent trigrams included to Language Model (LM). To account the influence of broader lexical context, the lattice might be rescored, i.e., a language model based on n-gram, $n > 3$, is re-applied for the decoded lattice.

In Postprocessor the result of decoding (or rescoring) is analyzed and transformed to one or more, in case of multi-decision, word sequence hypotheses supplemented with estimations of beginning, length, confidence and speaker identity for each word supplemented with restored punctuation as well as abbreviation and digital number representation.

D and KB parameters are estimated on speech and text corpora by means of training modules [30]. These modules allow for estimating parameters for models of speech patterns related to different Recognizer's components.

For punctuation restoration model parameters were trained on a large closed corpus. Evaluated models were build on publicly available data in order to keep the results obtained in this work reproducible.

The actualized speech-to-text conversion scheme made it possible to obtain the result of the broadcast recognition in a convenient appearance for perception and modifying by a human as well as for downstream automatic processing.

The developed web-interface allows for listening through and analyzing the speech signal synchronously with the raw text converted from speech, as shown in Figure 3. Here we can observe a bilingual segment where words lexically belonging to different languages are indicated with the character case.

Another transcription view allows for inspecting the generated punctuation marks synchronously with the media. Figure 4 illustrates how the restored punctuation, in general, facilitates perception of the extracted text. (Note that in this example first upper letter case is shown only after full stops and we preserve this notation in further explanation.) Three mistake types are gathered in the example illustration:

П'ЯТДЕСЯТИЙ ДЕНЬ ГОЛОДУВАННЯ ОЛЕГА СЕНЦОВА УКРАЇНСЬКОГО РЕЖИСЕРА ПОКИ НЕ ГОДУЮТЬ ПРИМУСОВО ПРО ЦЕ ЗАЯВИВ АДВОКАТ БРАНЦЯ КРЕМЛЯ ДМИТРО ДІНЗЕ ОДНАК ВІН НЕ ВИКЛЮЧАЄ ЩО ДО ТАКОГО КРОКУ РОСІЙСЬКІ ТЮРЕМНИКИ МОЖУТЬ ВДАТИСЯ ЯКЩО СТАН СЕНЦОВА ПОГІРШИТЬСЯ ВІН НИНІ ПЕРЕБУВАЄ У МЕДИЧНІЙ ЧАСТИНІ В'ЯЗНИЦЯ БІЛЯ ПОЛЯРНОГО КОЛА І НЕ МОЖЕ ХОДИТИ А ТИМ ЧАСОМ НА ПІДТРИМКУ УВ'ЯЗНЕНОГО РЕЖИСЕРОМ БАГАТЬОХ МІСТАХ УКРАЇНИ ТА СВІТУ ЛЮДИ ВИЙШЛИ НА АКЦІЇ НА ЙОГО ПІДТРИМКУ ГАННА БОКА ДАЛІ І ЙОГО МИ З ТОБОЮ мире мы с тобой ЛЬВІВ ДНІПРО ОДЕСА І ЩЕ З ДЕСЯТОК УКРАЇНСЬКИХ МІСТ

Figure 3: Raw result of speech-to-text conversion for a Ukrainian broadcast episode segment with indication of words from another language by the character case.

- The missing full stop between "…олега сенцова" ("…of oleh sentsov") and "українського режисера..." ("the ukrainian filmmaker…") can be explained by the proper name context that was unknown for the model.
- Comma substitutes a full stop between "…стан сенцова погіршиться" ("…sentsov's state will get worse") and "він нині перебуває…" ("he now is located…"), which still guides a reader to pause, disregard the wrong intonation modification.
- Between "…у медичній частині" ("…in the medical section") and "В'язниця…" ("The prison…") a full stop insertion is detected, which can be explained by the ASR-error caused substitution of the correct "у в'язниці" ("in the prison", pronounced as "u vjaznytsi") with incorrect "в'язниця" ("the prison", pronounced as "vjaznytsia").

In turn, the punctuation restoration was robust to the other error that might be interpreted as "…of many cities…", responded by ASR, instead of correct "…in many cities…".

So, it looks like, despite the mistakes, punctuation helps to transform voice data into more consumable form and to provide deeper level of understanding.

П'ятдесятий день голодування олега сенцова українського режисера поки не годують примусово. Про це заявив адвокат бранця кремля дмитро дінзе. Однак він не виключає, що до такого кроку російські тюремники можуть вдатися, якщо стан сенцова погіршиться, він нині перебуває у медичній частині. В'язниця біля полярного кола і не може ходити. А тим часом на підтримку ув'язненого режисером багатьох містах україни та світу люди вийшли на акції на його підтримку. Ганна бока далі і

Figure 4: Example of punctuation restoration in raw result of speech-to-text conversion for a Ukrainian broadcast episode segment.

Regarding the multilingual aspects, the presented transcribing system is able to simultaneously operate on two languages, that might be most often detected in Ukrainian broadcast, without any prior language detection. In practice, multilingual effects are expressed by speech, at least, in the following ways:

1. different speakers may speak switching to the language in accordance with the cultural or behavioral context;
2. speaker may confuse words in cross-lingual manner spontaneously sharing the same phoneme set for the confused word;
3. speaker pronounces a correct word in the context that is borrowed from another language, e.g., *включити* (to *include*) be used instead of *увімкнути* (to *switch on*);
4. speaker may use a version of the word adopted either phonetically or lexically from another language.

Cases 1-3 are covered by the speech-to-text conversion model, whereas the last, most complicated, case have not yet been considered, whereas, the punctuation restoration subsystem is supported in a single language recognition mode.

7. Conclusions

This paper presents a progress in narrowing the gap between a mere transcription and a more intelligent understanding of spoken language for Ukrainian.

While we have tested the performance of the presented punctuation restoration algorithm on texts with just cleaned punctuation, we have not yet measured the impact the speech recognizer's word error rate has on the F_1-score, a task we plan to address in the near future.

Dictionary reduction, which is crucial for highly inflective languages like Ukrainian, might benefit from subword automatic tokenization [31] as it experimentally proved to be constructive for certain ASR architectures [32].

In contrast to English, the considered training sample for Ukrainian is extremely small, so further corpus growing is one of main extra resources for improvement. From the other side, Brown-Uk corpus is publicly available and might be used to reproduce the results of this work and to compare them with other punctuation restoration systems.

Another prospective direction is the incorporation of prosody features. Its importance is stimulated by the fact that sentence constructions can be shared between different punctuation patterns that, in turn, are expressed with intonation cues. Although, a wider lexical context might be helpful as well, which is not always guaranteed to be presented. The latter is the common case for the online captioning task where accurate punctuation is extremely important in order to enable the audience to understand the context of the audio better.

Multilingual aspects are crucial for broadcast transcribing and Ukrainian broadcast provides input from two languages simultaneously [1], so the punctuation model is expected to support similar feature as well. A speaker diarization feature will be helpful for performance increasing as shown in [18].

Less frequent punctuation should be covered in future research and paired punctuation symbols like quotes and parentheses requires more sophisticated modeling.

8. Acknowledgements

This study was partially supported by National Academy of Science of Ukraine under the research project titled: "Development of Information Technology for Modeling of Spontaneous Speech Dialog Between Humans and Cybernetic Systems within Subject Areas", State Registration No 0118U000165.

9. References

[1] Sazhok, M., Seliukh, R., Fedoryn, D., Robeiko, V., Yukhymenko, O., Automatic Speech Recognition For Ukrainian Broadcast Media Transcribing, Control Systems and Computers. – № 6 (284), 2019, pp. 46–57.

[2] Sazhok, M., Robeiko, V., Seliukh, R., Fedoryn, D. and Yukhymenko, O, Written form extraction of spoken numeric sequences in speech-to-text conversion for Ukrainian. In CEUR Workshop Proceedings (2020) 2604, 442–451 (CEUR-WS, 2020).
[3] I. Kozlenko, Ukrainian punctuation: a handbook. “Kyyivskyy Universytet” Publishing Center, 2009 (in Ukrainian).
[4] Speechmatics, Why is punctuation important in speech recognition? https://www.speechmatics.com/blog/why-is-punctuation-important-in-speech-recognition
[5] W. Salloum, G. Finley, E. Edwards, M. Miller, and D. Suendermann-Oeft. Deep Learning for Punctuation Restoration in Medical Reports, in: Proc. of BioNLP 2017. doi: 10.18653/v1/W17-2319.
[6] X. Che, Ch. Wang, H. Yang, Ch. Meinel, Punctuation Prediction for Unsegmented Transcript Based on Word Vector, in: Proc. of Tenth International Conference on Language Resources and Evaluation (LREC 2016).
[7] Punctuation (in Ukrainian), http://pravila-uk-mova.com.ua/index/punktuacija/0-26.
[8] OnlineCorrector, https://onlinecorrector.com.ua/uk.
[9] LanguageTool, https://languagetool.org.
[10] Tilk, O., Alumäe, T.: LSTM for punctuation restoration in speech transcripts. In: Proceedings of Interspeech. pp. 683–687 (2015)
[11] O. Tilk, T. Alumäe, Bidirectional recurrent neural network with attention mechanism for punctuation restoration, in: Proceedings of Interspeech. pp. 3047–3051 (2016).
[12] TF-punctuator, https://github.com/dave-chatmost/TF-punctuator.
[13] X-Punctuator, https://github.com/kaituoxu/X-Punctuator.
[14] A. Oktem, M. Farrus and L. Wanner, Attentional Parallel RNNs for Generating Punctuation in Transcribed Speech, 5th International Conference on Statistical Language and Speech Processing SLSP 2017.
[15] S. Peitz, M. Freitag, A. Mauser, and H. Ney, “Modeling punctuation prediction as machine translation.” in IWSLT, 2011, pp. 238–245.
[16] Farrús, M., Lai, C., Moore, J.D.: Paragraph-based Prosodic Cues for Speech Synthesis Applications. In: Proceedings of the 8th International Conference on Speech Prosody (2016).
[17] Live captioning in Google Meet expanding to 4 new languages, https://cloud.google.com/blog/products/google-meet/live-captioning-in-google-meet-expanding-to-4-new-languages.
[18] Hlubík, P. & Španěl, M. & Boháč, M. & Weingartová, L., Inserting Punctuation to ASR Output in a Real-Time Production Environment, in Proceedings of 23rd International Conference, TSD 2020, pp. 418–425.
[19] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N.A., Transition-based dependency parsing with stack long short-term memory. CoRR abs/1505.08075 (2015), http://arxiv.org/abs/1505.08075.
[20] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” ICLR2015, arXiv:1409.0473, 2015.
[21] T. Wang and K. Cho, Larger-context language modelling, arXiv preprint arXiv:1511.03729, 2015.
[22] 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.
[23] Brown-Uk: Contemporary Ukrainian Language Corpus, https://github.com/brown-uk/corpus/tree/master/data
[24] S. Matveyeva. Tokenization as a way to corpus text processing, in: Proc of I International Conference “Applied and Corpus Linguistics: New Generation Technology Development”, 2018, pp. 37–38 (In Ukrainian).
[25] Lang-Uk community homepage, https://lang.org.ua/en.
[26] FastText word embeddings, https://fasttext.cc/docs/en/crawl-vectors.html
[27] Zheng-Hua Tan, Achintya kr. Sarkar and Najim Dehak, “rVAD: An Unsupervised Segment-Based Robust Voice Activity Detection Method,” Computer Speech and Language, 2019.

[28] Najim Dehak, Patrick Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, Front-End Factor Analysis for Speaker Verification, in: IEEE Transactions on Audio, Speech, and Language Processing, 19(4), pp 788–798, 2011.
[29] A. W. Zewoudie, J. Luque, J. Hernando. The use of long-term features for GMM- and i-vector-based speaker diarization systems, EURASIP Journal on Audio, Speech, and Music Processing (2018) 2018:14.
[30] Povey D., Ghoshal A., Boulianne G. et. al., The Kaldi Speech Recognition Toolkit, IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.
[31] T. Kudo, J. Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, in: Proceedings of EMNLP2018.
[32] Hayashi, T., Yamamoto, R., Inoue K. et al., Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2020), pp. 7654–7658.