Producing a Machine Learning System to Detect Word and Sentence Boundaries and Restore Spaces, Punctuation and Capitalisation in Lowercased Non-Whitespaced English Text

Travis Browning*, Moblessing Mudzamiri*, Natalie Philipps*, Christopher Vidler*
Research Group in Computational Linguistics
University of Wolverhampton
Wolverhampton, United Kingdom
{t.s.browning, m.a.moblessing, n.a.philipps, c.s.vidler}@wlv.ac.uk

***Abstract*—In this paper, we develop components for a machine learning model that is designed to detect word and sentence boundaries, restore periods and commas and restore capitalisation in a piece of lowercase English text that is devoid of whitespace and punctuation. Our model pipeline consists of a statistical and dictionary-based space restoration component; a punctuation restoration component that uses a bidirectional LSTM based on an existing solution (Punctuator2); a rule-based sentence splitter; and a capitalisation restoration component based on an existing solution (Truecaser). Though we are unable to test the full pipeline, we nonetheless obtain interesting results from testing the individual components, including observations of the linguistic feature most easily recognised by our component for restoring punctuation (namely, the presence of a sentence-ending period in the vicinity of a marker of direct or reported speech such as "he said"), and neologisms such as "self-isolate" that posed an issue for our capitalisation restoration component in light of the data that it was trained on.**

Keywords—whitespace, SBD, punctuator, punctuation truecaser, capitalisation, BiLSTM

I. Introduction

As part of the end-of-module assessment for the Machine Learning module (7LN008) on the 2020-21 University of Wolverhampton MA Computational Linguistics programme, students taking the module were divided into groups and asked to develop a machine learning solution in the course of a groupwork task. This report contains the results of our group's work on this assignment.

The brief for the assignment was to build a machine learning system that would detect word boundaries (i.e. restore spaces in between words), detect sentence boundaries, restore punctuation (limited to periods[1] and commas) and restore capitalisation in a piece of English text that is devoid of whitespace and punctuation and consists solely of lowercase alphanumeric characters. As a visual example of what our system sets out to achieve, the envisaged ideal system would take a non-spaced lowercase string such as "thisstringhereafterbeingputthroughoursystemwouldlookmorelikeapropersentence" and return it as "This string here, after being put through our system, would look more like a proper sentence."

Where possible, we have tried to come up with original solutions for the components of our model and draw on the algorithms and approaches that we have been taught in the Machine Learning module, as well as other natural language processing skills that we have acquired over the course of our degree programme. Nevertheless, we have occasionally incorporated existing published solutions that we have found in our review of related work, either for reasons of reducing model complexity (particularly in light of time constraints), or because we felt that these solutions already solved the specific sub-task particularly well and it would be difficult to significantly improve on these. We extend our gratitude here to the authors of code sections and solutions that we have incorporated into our model or that otherwise served as inspiration for our work.

Finally, as you will see in our results and conclusions, significant issues do remain with our model. Indeed, we did not expect to be able to create a model that would restore spaces, punctuation and capitalisation perfectly; this is in part due to the complexities of the English language, where there is great potential for ambiguity in interpreting parts of speech, for example. Nevertheless, our work still yielded some interesting results from a linguistic perspective, particularly when examining the output in detail for patterns and trends, and we are glad to be able to share these findings with you here.

II. Related Work

Our original intention was for our model to sequentially follow the four stages outlined in the brief (restore spaces > identify sentence boundaries > restore punctuation > restore capitalisation). Below, we present a summary of our research on related work to understand more about the existing solutions and strategies for completing each of these tasks.

A. Restoring Spaces

Written language that features no whitespace between words, or *scriptio continua* (continuous writing), is not uncommon, since this is not only a feature of many contemporary east Eurasian languages – including Chinese, Thai and Japanese – but in the past was also the most common and official way of manifesting written Latin, Greek and other languages (Mohr [1]). This writing style has much potential to cause problems for automatic translators, which, having been designed primarily by native speakers of languages that use whitespace to separate words, typically rely on this whitespace to identify word boundaries. Thus, the process of separating

[1] *Although this report is written in British English, "period" has been used in place of "full stop" throughout for simplicity and brevity.*

** Students on the University of Wolverhampton MA Computational Linguistics programme (2020-21).*
Authors ordered here alphabetically – for specific details on each member's contribution, please see the "Acknowledgements" section.

words with whitespace, known as word segmentation, is critical to master. In principle, to add spaces between words, the continuous text must be scanned for all potential word boundaries; the most likely sequence of words to form the complete sentence is then chosen and the spaces inserted accordingly. In practice, however, this initially daunting task becomes more manageable with clever modelling.

Perhaps the most common approach to word segmentation is to employ conditional random fields (CRFs) [1]. Since these analyse samples not in isolation but in relation to nearby samples, CRFs are thus able to pay attention to the order in which words appear, and how they are able to influence each other, altering aspects ranging from the semantic meaning of the words, such as in sense disambiguation, to the grammatical possibility and probability of two words appearing next to one another in a sentence (Zhao, Huang and Li [2]). In other words, CRFs can indicate the likelihood of two potential words being next to one another in a sea of other potential words.

Maximum entropy is another commonly employed model for word segmentation of *scriptio continua*. This trains off of data that has been tagged for parts of speech (POS), meaning that words have been located, which in turn means that certain characters can then be tagged as the left boundary of a word, the right boundary, or within a word (Xue [3]). The entropy model then attempts to construct words from the sequence of characters, giving them POS and word-location tags as per the data it was trained on, as well as certain semantic concepts like plurality, to find a likely sequence [3].

Statistical and dictionary-based approaches can also be used for word segmentation [3]. This relies on probabilities that any known word will appear in a text and uses a finite-state transducer to construct probable strings on the basis of these probabililites [3]. Maximum-matching algorithms are especially non-preferable; these essentially construct words that are large as possible on the basis of their training data, which works well enough for Chinese, though still struggles, particularly with ambiguity [3]. Regardless, this approach works significantly less well for English.

In the case of Chinese word segmentation carried out by Zhao, Huang and Li [2], both CRFs and maximum entropy turn the problem into that of label-tagging, though CRFs allow smaller tagsets for the same data, with said tags denoting the position of a character in a potential Chinese word [2]. Chinese words are typically smaller than English, consisting of only about four characters as a maximum [3], making these approaches relatively cheap. Both return results with precision and recall scores of higher than 94% [2] [3]. Regardless of the model opted for, the method is only as effective as its training; although Zhao, Huang and Li [2] found their different models to be rather equally effective at finding the same word boundaries, they relied on different models that were trained on different corpora to fill the gaps of unfound word boundaries. However, using multiple models was found to be extremely taxing on memory and use of time when used with CRFs [2].

The task of handling English words in *scriptio continua*, however, makes the use of maximum entropy harder, as tagging individual characters of words to demarcate their boundaries is less useful, given the smaller number of characters in the English alphabet and the greater length of English words. Employing statistical and dictionary-based approaches were integral to the model we ended up employing, with its frequency dictionary defined by the Reuters articles that constituted our training data. It seemed important to integrate statistical and dictionary-based ideas, given the specific lexicon used in news media. Zipf's law was useful in determining a word's likelihood in our frequency dictionary so as not to make rare words appear too often.

B. Identifying Sentence Boundaries

With regard to the second stage of the system (detecting sentence boundaries in English text with spaced words, but without punctuation or capitalisation), this proved to be one of the most challenging areas to find examples of related work on. This is due to the inherent fact that detecting sentence boundaries in non-punctuated text is not a common problem faced when processing English text, which is normally obtained in an already punctuated and capitalised form.

Consequently, the vast majority of existing sentence boundary detection (SBD) systems rely on some form of punctuation being present to perform effectively. Commonly cited examples of early systems include the Satz system [4], and mxTerminator [5], which uses a maximum entropy approach. Other prominent systems that were subsequently developed include Punkt [6] and splitta [7]. Comparisons of such systems and the approaches that they use have been carried out by Walker et al. [8] and Read et al. [9], while a more recent overview by Griffis et al. [10] compares the performance of five of these systems on texts specific to the medical domain.

Since virtually all of these systems were designed to process already punctuated text, however (e.g. to resolve sentence boundary ambiguities posed by the presence of periods in abbreviations), it felt necessary to look to other areas of research that aligned more closely with the type of input text that our system needed to process at this stage. One application that seemed ideal in this regard is the processing of transcribed speech, since this involves identifying grammatical structures and sentence boundaries that in text that is naturally initially unpunctuated. Relevant research in this area includes a study by Stevenson and Gaizauskas [11] that explores the use of the Timbl memory-based learning algorithm to delineate sentences in transcriptions produced by automatic speech recognition (ASR) systems, as well as an approach by Liu et al. [12] using a conditional random fields model, which was found to yield a lower error rate than previous approaches that used a hidden Markov model and maximum entropy classifiers.

We were able to find an out-of-the-box solution that appeared to be perfectly tailored to the requirements of this stage of our system: a sentence segmentation system called DeepSegment developed by Bedapudi [13] that was specifically designed to work on non-punctuated (or poorly punctuated) text. DeepSegment's initial implementation uses a bidirectional LSTM-CRF model in conjunction with GloVe for word representations. This was compared to another neural network system called NNSplit developed by Minixhofer [14], which was also designed to be less reliant on aspects such as punctuation and case. Ad hoc testing carried out on these systems suggested DeepSegment to be a slightly more reliable choice for incorporation into our model than NNSplit, with the latter system splitting our test texts more frequently to create a number of incorrect breaks, interpreting single words and phrases such as "however" or "on 21 june" as standalone sentences. Ultimately, though, as described in the

methodology section that follows, we decided to forego the inclusion of a separate sentence boundary detection component in favour of a system that would directly insert punctuation into the text after the space restoration stage, without the need to delineate sentences first.

C. Restoring Punctuation

Automatic punctuation restoration is a common post-processing problem in natural language processing (NLP). Studies in this area are often linked to the issues of sentence boundary detection highlighted above and many of them therefore also focus heavily on restoration in ASR systems, since these systems can produce text data that is unpunctuated and difficult for humans to interpret. Readability is important not only for human readers, but also for other NLP tasks. Therefore, punctuation restoration systems should produce speech data that is ideal for this purpose.

There are several studies on automatic punctuation restoration systems. Literature shows that punctuation restoration systems are most commonly applied in ASR (Tilk and Alumäe [15]; Salloum et al. [16]; Alam, Khan and Alam [17]; Augustyniak et al. [18]).

To produce a well-functioning punctuation restoration model for text data models or ASR models, the preprocessed data used to train the model must be evaluated for errors as much as possible. This is because errors that may not be present in the training data but are present in the test data may degrade the performance of the model. The lack of or incorrect placement of punctuation are inaccuracies that can potentially change the meaning of a sentence, resulting in ambiguity. Therefore, it is essential to mitigate ambiguity when dealing with text data, in order to avoid training this ambiguity into machine learning algorithms.

Vocabularies that contain a large number of domain-specific words can impose difficulties on creating well-performing punctuation restoration models for the general domain. A study on punctuation restoration in medical diction data (Salloum et al. [16]) found that domain-specific jargon increased issues related to data sparsity and out-of-vocabulary (OOV) words. A low-parameter model is ideal for quickly decoding speech in real time. To address these issues and reduce the size of the vocabulary, Salloum et al. [16] performed preprocessing steps such as normalising all the OOV words in the text: for example, the integers in dates were replaced by the letter "D". They also harmonised abbreviations and converted these to lowercase. However, state-of-the-art methods such as bidirectional recurrent neural networks (BRNNs) require large amounts of data to give the best output.

On the other hand, another approach to punctuation restoration using neural networks is word tagging. Our strategy follows a similar approach of using data tagged with punctuation markers, for which we substituted commas and periods with the tags ",COMMA" and ".PERIOD". The trained model should be able to identify where to place commas and periods in place of the trained tags. Augustyniak et al. [18] follow a similar strategy; however, they include markers for other forms of punctuation such as exclamation marks, question marks and ellipses, while the remaining punctuation markers were trained to be left as whitespace due to their low frequency. Other studies have also chosen not to identify and tag all punctuation: for example, Gravano, Jansche and Bacchiani [19] found that commas and periods accounted for 40–47% of punctuation in their dataset. So there is still a great deal of significance in only focusing on restoring commas and periods, as per the assignment brief.

Inspired by Tilk and Alumäe [15], we built a model with a BRNN, which is constructed using two separate recurrent neural networks that iterate forwards and backwards through all the available input data. This method gives us information from the past and future of a specific time frame (Schuster, Kuldip and Paliwal [20]). The bidirectional layer in the model allows us to keep track of the relevant context of the words that come before and after the current position in the text.

The model also has attention mechanisms (Bahdanau, Cho and Bengio [21]) which are used to take advantage of the long-range dependencies to the left and the right of each word. They encourage the model to find relevant tokens of text to make punctuation placement decisions. For example, when placing periods to mark the end of a sentence, the model will focus on finding words that indicate periods. However, due to the distance from the current word, the model will need to be encouraged to decide on sentences which require periods as opposed to question marks, for example.

The combination of the BRNN and attention mechanism gives a model the advantage of being able to utilise the context of words in both directions and a focused attention at different time stamps. This should hopefully enable our model to make more decisions that are correct when restoring punctuation.

D. Restoring Capitalisation

As the digital world expands, so does the influx of low-quality text data. Transcripts of text data can be constructed from various sources such as ASR, online messaging, emailing, gaming, web searches and many other examples (Lita et al. [22]). Unfortunately, these sources often contain misspellings, grammatical errors, insertions, and many other elements that reduce the quality of our data.

The process of restoring the correct capitalisation to input data that does not contain any case information or is incorrectly cased is also known as "truecasing". It is a beneficial process for tasks such as machine translation (MT), named entity recognition (NER) (Grishman and Sundheim [23]) and content extraction, since the correct capitalisation of words can lead to better accuracy.

Early truecasing studies approached capitalisation restoration as a sequence tagging problem. For example, Huang and Zweig [24] develop a maximum entropy-based strategy for the annotation of spontaneous conversational speech. They insert punctuation as tags using prosodic and lexical features, intending to make automated transcriptions more readable. Following this approach, Kim and Woodland [25] conduct a similar tagging strategy that focuses on tagging capitalisation on named entities in the speech input.

Another common approach is using *n*-gram models from a corpus that already contains case information. For example, Gravano, Jansche and Bacchiani [19] investigate truecasing using *n*-gram models. The impact of scaling the size of the corpora and increasing the *n*-gram order was investigated with the intention of creating a model that restores punctuation and capitalisation in a single pass. The dataset collected from news articles from the internet was split, and various *n*-gram orders from $n=3$ to $n=6$ were tested for efficiency. The inspiration to view punctuation restoration and truecasing as a combined task came from the fact that some capitalisation decisions are

determined by inserting punctuation marks that indicate sentence boundaries.

Much like punctuation restoration, many of the studies that have been conducted on truecasing have a strong focus on ASR systems. For example, Rei, Guerrerio and Bastista [26] investigate using the BERT model (Devlin et al. [27]) to automatically recover capitalisation on video subtitles.

Since most of the methods mentioned above cater to ASR, it felt appropriate to find methods that are more aligned with the Reuters-21578 news dataset that we ultimately chose to use in this work. Lita et al. [22] provide an early example of a truecasing system trained on news articles. They propose a language model that uses trigrams to capture the local context in texts and then to bootstrap this context across the sentences in the text. Their results show a 98% accuracy performance on assigning the correct case labels to the tokens.

III. Dataset

For training our model, we chose to use the Reuters-21578 dataset, one of the most widely used corpora for text categorisation research that was originally collected by Carnegie Group, Inc. and Reuters Ltd. The dataset is comprised of 21,578 English language Reuters newswire stories from 1987, hence the name. The collection is distributed into 22 files, each containing 1000 documents, bar the last file which contains 578 documents. There are five sets of content-related categories which were classified manually by a human who decided on the placement of each document.

The collection of files is in standard generalized markup language (SGML) and therefore has SGML tags that divide each file, document, and section. For this study, we trained and tested our model using ten documents from the 22 SGML files. We extracted the data between the <BODY> tags to retrieve the main texts of the stories.

IV. Methodology

Our initial intention was to develop the model according to the four stages outlined in the related work section above: i.e. to build a model that first restores spaces, then detects sentence boundaries from among the spaced words, then inserts commas and periods in each sentence, then finally capitalises the necessary characters in each sentence.

However, our approach changed over the course of working on the project. We realised that it may be simpler to train a model to insert punctuation directly into the continuous (i.e. non-sentence-delineated) spaced text without having to determine the sentence boundaries first, effectively completing stages 2 and 3 in a single step. A sentence splitting stage was nevertheless added after the punctuation stage to assist the model for the capitalisation stage – like many of the existing sentence splitters identified in the related work, this relies on already inserted sentence-ending punctuation as a basis for splitting the sentences. This is a simple rule-based step that does not rely on machine learning techniques; however, there is still potential for further improvement in this regard. For example, this approach currently assumes that all inserted periods are sentence-ending – to resolve this, a classifier that uses an algorithm such as k-nearest neighbours (kNN) could be trained here to detect whether a period is sentence-ending or actually forms part of an abbreviation (e.g. "U.S.").

Our final model thus takes the following structure: a statistical and dictionary-based space restoration component; followed by a punctuation restoration component that uses a bidirectional LSTM, which is based on the Punctuator2 system from Tilk [15]; followed by a rule-based sentence splitter; followed by a capitalisation restoration component, which is based on the Truecaser system from Reimers [28].

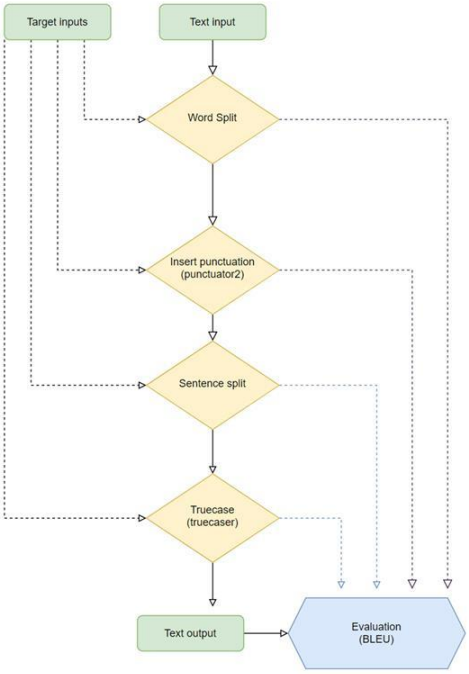

Fig. 1 - A flowchart diagram of our model's structure.

A. Restoring Spaces

Ultimately, the model settled on for word segmentation was statistical and dictionary-based. This solution would be largely inspired from a response provided by an anonymous user in response to a question asked on a Stack Overflow thread ("How to split text without spaces into list of words" [29]). The text would be parsed for all potential words found within the dictionary, and a cost would be assigned to each potential sequence of words based on the frequency of the words within each potential sequence. Whichever sequence had the lowest overall cost would be selected as the most likely sequence, and spaces would be inserted into the string between each of these determined words.

To obtain the frequency dictionary from the training data, a copy of the news was rendered into a string and scanned, adding a count to a dictionary every time a word was encountered. This dictionary was then sorted and rendered back into a string with line breaks separating the words, now sorted by most likely to least likely. Using Zipf's law, scores would be assigned to each word in the frequency list to maximise the likelihood of more frequent words without rendering the less frequent ones impossibly unlikely.

This was tested against many other Reuters news articles, which were prepared for testing by having all of their punctuation and spacing trimmed, rendering them as English scriptio continua. Saved as a string, it was fed to the method, which output results largely matching those of the original text.

Fig. 2 - The first image (top) is the original text. The second image (bottom) is the text produced by the algorithm.

As a linear algorithm, it is fairly fast and efficient, with total processing taking less than five minutes on the device it was tested on. For very large strings, though, a lot of memory can be consumed to process them. In the case of memory overexpenditure, the input string (the text whose spaces will be restored) could be split into multiple sub-strings, but this is very risky if the input is already rendered as *scriptio continua*, as words and sentences might well be split apart. Thus, given this possibility, this approach is not recommended.

One known flaw of the model is its inability to handle unknown words, i.e. those that are not in the trained dictionary. Words that were coined after the training files were created are going to be missed, and thus it may be wiser to use training text that is more up to date when training future implementations of the model. Whenever such uncertainties are encountered, the function finds smaller, nonsensical subwords as the best fit, which sometimes take the form of individual letters. This has the risk of causing the method to essentially "give up" for the rest of the text, adding a space after every single letter. However, in final testing, this phenomenon thankfully did not occur: it was most strongly associated with the use of limited training data at the start. Another issue with the method is that it does assume that words are relatively independent, thus the probability of any one word being chosen is not as influenced by its surrounding words as it should be. Stronger algorithms, particularly CRFs, would overcome this issue. Future design with this project would see the proper implementation of one of these algorithms.

B. Restoring Punctuation (Punctuator2)/Sentence Splitting

The original approach to sentence segmentation anticipated segmenting text without punctuation. Implementations including Deep-EOS [30] and DeepSegment [13] were attempted. However, the antiquated state of the packages required to run these pieces of software proved to be an ever-present challenge.

Ultimately, it was decided to address the punctuation of words first, before segmenting the text into sentences. An implementation of BertPunc [31] was attempted for punctuation restoration; however, it appeared that the training data, which was annotated to read like the example training data, would not be accepted by the preprocessing step. Due to the accuracy that BertPunc reports, it would be worth exploring recreating the training data to be read by the preprocessor in a future implementation.

Punctuator2 [15] (GitHub: [32]), a bidirectional long short-term memory (BiLSTM) architecture, was decided upon as the punctuation restoration model that would be used, as the comprehensive documentation was straightforward and easy to follow. Detailed in README.md, the implementation includes preprocessed and cleaned data placed in a directory, and a model can be finetuned in Windows through CMD using two lines. Our model, which was trained through CMD on 15,094 lines of training data, was tested on 4302 lines of data and validated with 2166 lines of data. A recommended hidden layer size of 256 and a recommended learning rate of 0.02 were selected for our model. This finetuning was performed on an i7-8550 CPU at 1.80 Ghz, with 8 logical cores and 16 GB of memory, over the course of 5 epochs in approximately 4 hours.

Once a model was trained, punctuation restoration was handled in punctuator.py through a shell script in Google Colaboratory ("Colab") in under a minute. Punctuator2 does include error_calculator.py for evaluation purposes. However, this package was disregarded in favour of using the BLEU score at the end of the pipeline.

Output texts from Punctuator2 are post-processed to remove punctuation that is printed to resemble labelled tags in the training data (".PERIOD", ",COMMA") and replace these tags with the punctuation marks on their own. This text is then read on a string level for word-final occurrences of period characters. Words are added to a blank string as a sentence. If this character appears as the last character in a word, the word is added to the string, the string is appended into a list of strings, and the sentence is reset as an empty string. Though this process will work for most sentences, it is an opportunity to improve our output. For example, this sentence segmenter will fail every sentence that contains a multi-word abbreviation such as "U.S.A." or "U.K.". A word-level kNN regressor trained on trigram distributions is predicted to have a positive effect on the evaluation scores compared to this model.

C. Restoring Capitalisation (Truecaser)

Case restoration for text without case information is a widely explored field of natural language processing. Outside of machine learning environments, packages such as the Natural Language Toolkit (NLTK) [32] contain part-of-speech taggers, such as Punkt [6], that are capable of handling basic named entity recognition tasks on punctuated text. More accurate named entity recognition packages, such as Stanza [33], would be able to perform truecasing with relative accuracy when paired with character-level string editing techniques. After preliminary testing, Stanza was found to struggle with multi-word abbreviated names, such as "U.K." or "U.S.A.". Although such entities were clearly identified by Stanza, their treatment would be similar to other named entities, resulting in an uppercase output of the first character. This minor issue could be addressed in post-editing, although an output that circumvents this altogether would be more ideal.

The most effective output explored was inspired by the statistical model described in Lita et al. [22]. Reimers' Truecaser [28] regressively evaluates tokens on a trigram level

to assess their statistical likelihood of casing formats. TrainTruecaser.py was used to train an object on the first 151 characters of every article. In initial testing, this distribution, "d1.obj", performed similarly to objects trained using corpora from NLTK, as well as the provided object, "distributions.obj").

Object Name	Included Training Material	No. of Lines (len)	% of Correct Tokens
distributions.obj	nltk.brown.sents()	57340	93.16
d1.obj	train0.txt [s[:150] for s in reuters]	9191	90.29
d3.obj	nltk.gutenberg.sents()	98552	93.23
d4.obj	Brown + Gutenberg	155892	93.02
d5.obj	train0 + Brown + Gutenberg	165083	93.23
d6.obj	(Repeat of training material for d5.obj)	165083	93.16

Fig. 3 - Truecasing performance by object on example test data.

The performance of d1.obj was not far behind the performance of the other models, and was a component of the highest performing model, which was composed of NLTK's Brown and Gutenberg corpora as well as our subsection of Reuters.

In terms of implementation, Truecaser was a straightforward build. The package was originally written for Python 2.7, with a bit of legacy code remaining in the current release. The main obstacle in implementing Truecaser was editing the Truecaser to open our various outputs and test outputs for processing.

Based on self-evaluation metrics, Truecaser appears to be an effective solution for restoration of case in text without case information.

V. Evaluation and Results

A. Evaluation Strategy and Metrics

Evaluation was handled by the Bilingual Evaluation Understudy (BLEU) [34] score, as provided by NLTK. Firstly, target outputs are generated by processing a text, in order to produce goals for every individual step in the pipeline. Individual steps are fed the previous step's target output to produce a step-level output for evaluation. In addition to this, a text that has been processed straight through the entire pipeline is compared to the final target text.

As evaluation had not been planned for from the beginning of the process, inconsistent file handling behaviours over the course of the first run of the experiment lead to wildly inconsistent BLEU scores when handled at the sentence level. This issue can be resolved by defining file handling procedures at the beginning of the next implementation.

Going forwards, it will be worth exploring other evaluation options. Intento-API's score.md [35] appears to include evaluation metrics such as ROUGE and RIBES, as well as multiple BLEU scores.

B. Preliminary Evaluation of Individual Components

In order to have some preliminary data on our model's performance to discuss for the purposes of this report and its accompanying presentation, the Punctuator and combined sentence splitter and Truecaser components of the model were tested individually, using the ideal output that would be produced by the preceding stage as input in each case. The output from these individual tests can be viewed in Annex A and Annex C for the Punctuator and Truecaser components respectively.

Numerical data was obtained for the performance of the Punctuator component, in order to calculate the precision, recall and F-measures for each punctuation mark in a given test text. To evaluate this component's performance, its output was manually annotated according to a scheme as shown in Annex B to indicate where the model correctly or incorrectly added a comma or period, or where these punctuation marks were missed and should have been added.

When tested on a sample news article of approximately 880 words, the model was able to restore 5 periods and 2 commas in the correct position that they appeared in the original article. 2 periods and 2 commas were incorrectly inserted in positions where they did not feature in the original article. Naturally, given the size of the text extract, a great deal of punctuation was also missed: no punctuation marks were restored in a total of 78 positions where punctuation marks should have been, amounting to 43 periods (36 sentence-ending, 7 from abbreviations such as "U.S."/"Man.") and 35 commas that were present in the original article. It should be noted that one of the correctly restored periods was the final period inserted at the very end of the test extract.

These results are summarised in the following table, along with the precision, recall and F-measure scores calculated to three decimal places:

	Correct	Incorrect	Missed	Precision	Recall	F_1
Comma	2	2	35	0.500	0.054	0.098
Period	5	2	43	0.714	0.104	0.180

Fig. 4 - Table showing the obtained counts and scores for correct, incorrect and missed commas and periods in the output shown in Annex A, as per the annotations in Annex B.

From the results obtained above, it seems naturally evident that the Punctuator component of our model experiences significant issues with recall, and would likely benefit from further fine-tuning to improve its inclination to insert punctuation marks. Nevertheless, it is still noteworthy that this component of our model has yielded a relatively high level of precision in correctly restoring periods, even if this concerns a very small data sample. Further tests would ideally need to be carried out to verify that the model can continue to replicate such scores.

Possible reasons for the model's high level of precision in restoring periods in this case are discussed in further detail in the linguistic analysis section below, along with a detailed assessment of the Truecaser output.

C. Linguistic Analysis of Preliminary Results

To understand more about how our model's individual components were performing, we carried out some linguistic analysis on the preliminary output obtained from the Punctuator and Truecaser components as discussed in subsection B immediately above. As noted there, the output from these individual tests can be viewed in Annex A and Annex C for the Punctuator and Truecaser components

respectively, with the manually annotated output for the Punctuator shown in Annex B. We were able to obtain a few intriguing insights from this analysis, which are discussed in more detail here.

One of the most interesting things to note is that of the seven periods that our Punctuator component decided to insert (six when discounting the final period inserted at the very end of the extract), four of these periods were either directly preceded or followed by a marker of reported or direct speech ("he said", "she also said", "she said", "saunders said"). This suggests that one of the key features the model component recognised in training was that such speech markers commonly begin or end a sentence and often require a period after "said" or before the corresponding noun or pronoun. However, it is notable that there are a number of other instances of "said" in our test text preceded by both pronouns and proper names where this phenomenon did not occur.

Of these four instances where a period was inserted either before or after a speech marker, two were inserted in the correct position according to the punctuation in the original text, while two were inserted the wrong "side" of the speech marker, i.e. following the "said" rather than preceding the pronoun/proper name as per the original text. It is also notable that these were the only two periods that the model component inserted incorrectly, even if the model component still missed a large number of periods that were present in the original text.

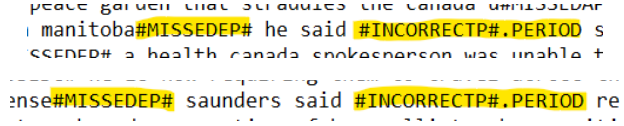

Fig. 5 - Two instances where a period was incorrectly inserted after "said" rather than before the corresponding pronoun/noun as per the original text. #MISSEDEP# indicates a sentence-ending period present in the original text that was missed, #INCORRECTP# indicates an incorrectly placed period.

Indeed, since these speech markers could feasibly either begin or end a sentence, this likely poses a key issue for a model when deciding whether a period should be placed before or after a speech marker of this kind. This ties in with our observations when testing the DeepSegment and NNSplit systems created by Bedapudi [13] and Minixhofer [14] respectively, where we noticed that both systems similarly struggled with whether to split a sentence before or after these markers and effectively had a 50% chance of getting this right.

The other preliminary output evaluation was carried out on the Truecaser component, the output of which is shown in Annex C. The results here were not annotated for error counts in the same way as for the Punctuator component output, but a cursory examination of the resulting text shows that the Truecaser appears to have correctly restored capitalisation in most cases, particularly for named entities. It does appear to have a tendency to err on the side of initially capitalising the word in cases of uncertainty, however, and has occasionally incorrectly capitalised mid-sentence verbs such as "opt" and "prioritizing". One aspect worth noting is the contextual nature of the piece: the news article used for testing relates to the COVID-19 pandemic and features words such as "self-isolate", which would not have appeared in our training data, and it is perhaps not a surprise that our model component chose to capitalise this. However, words such as "vaccinate(d)" and "quarantine" were curiously also capitalised, which are not neologisms specifically related to COVID-19 and so theoretically should have been recognised as existing verbs/nouns that did not require capitalisation by the model component.

The other observation of note is that, since our model was not required to restore other forms of punctuation such as apostrophes, contractions such as "couldn't", "hasn't" and "wasn't", meaning that these were rendered as "couldnt", "hasnt", and "wasnt" in the input that was fed to the component. The Truecaser has been unable to recognise these as contractions without the apostrophes and so has added an initial capital to all of these where they appear mid-sentence.

VI. Conclusions

Although time constraints and other difficulties have prevented us from developing our model to the full standard of completion that we desired, we have nevertheless been able to produce working prototypes for all its individual components and also obtained some insightful data in the course of testing these. Individual analysis of our model's components shows areas for improvement, but other areas in which the components are performing to expectations.

Our space restoration component struggles with unknown words and initially tended towards breaking continuous character groups down into single letters separated by whitespace, yet improved considerably after being trained further with additional data, and could be improved further still with the use of algorithms such as CRFs. Our combined SBD and punctuation restoration component currently has issues with recall and outputs a mix of a tagged and non-tagged punctuation, yet the few instances of restored punctuation were relatively precise – particularly with regard to periods, where we were also able to obtain some insight into the features that our model was able to most easily able to learn from the training data, namely the common occurrence of a sentence-ending period in the immediate vicinity of a speech marker such as "he said". The SBD sub-component could notably be improved further to avoid segmenting on abbreviations, e.g. through the use of a kNN algorithm. Finally, our capitalisation restoration component appeared to perform remarkably well, though it appears to have a tendency to over-capitalise in the case of uncertainties and was shown to struggle with out-of-vocabulary words and contractions that lack apostrophes.

There is naturally still scope for further work and improvement on this project, and we regret that we were unable to evaluate the model in full as a complete pipeline; however, the results from evaluation of the individual components prove relatively promising and go some way towards fulfilling the brief for this task.

Acknowledgments

The authors would like to thank Dr Burcu Can Buğlalilar (Research Group in Computational Linguistics, University of Wolverhampton) for all her help in this task, as well as all other students on the 2020-21 MA Computational Linguistics programme for their general support and feedback.

The authors are also grateful to one another for their mutual support and dedication in completing this groupwork.

With regard to the individual contributions from each member and our strategy for approaching the group task, our initial approach was to divide the workload for the envisaged four-stage model equally among the group members, with

each member tasked to review related work for, research existing solutions for and come up with initial propositions for their respective component of the model. The components were assigned as follows:

- Philipps: Stage 1 (Restoration of Spaces)
- Vidler: Stage 2 (Sentence Boundary Detection)
- Mudzamiri: Stage 3 (Restoration of Punctuation)
- Browning: Stage 4 (Restoration of Capitalisation)

Browning, Mudzamiri and Vidler also jointly researched and selected the dataset used to train our model at an early stage of the process.

As the work progressed, however, and due to changes in the structure of our model, we shifted towards a strategy where group members focused on their individual strengths in completing the remainder of the work, assuming responsibility for a certain aspect of the project. Philipps continued to focus on code compilation for the first stage (space restoration) of the model. Browning took charge of compiling the code for the remaining stages of the model (punctuation and capitalisation restoration) and also provided ample assistance with regard to troubleshooting, debugging and computational issues. Browning also proposed strategies for evaluating the model as described in the "Planned Evaluation Strategy and Metrics" subsection of this report.

Mudzamiri and Vidler shifted their focus to evaluation ahead of the planned presentation for our work, with Mudzamiri proposing strategies for evaluation and Vidler performing preliminary evaluation and linguistic analysis on the output of individual tests of the Punctuator and Truecaser components. Throughout the course of the project, Mudzamiri also assumed responsibility for organisation of group meetings and set scheduled targets for the group to work towards. Vidler assumed responsibility for the compilation, editing and proofreading of the final report, as well as for the design and coordination of the accompanying presentation that was given on 7th June 2021 to conclude a series of lectures for the Machine Learning module.

All group members have contributed written sections to this report, which mostly correspond to their area(s) of contribution as described above. Mudzamiri took over the write-up for the related work sections on punctuation and capitalisation restoration, and also contributed the section that provides information about our dataset. Philipps provided the related work and methodology sections for the space restoration component. Browning contributed the methodology sections for the punctuation restoration (Punctuator), sentence splitter and capitalisation restoration (Truecaser) components, and also contributed the subsection on planned evaluation strategy and metrics for the overall model. Vidler contributed the related work section on sentence boundary detection and the subsections on the preliminary evaluation of the individual components and the linguistic analysis of these results, as well as all remaining sections of this report.

Although each group member's individual contributions and areas of responsibility have been outlined in detail above, it must be emphasised that the authors continued to help each other out with their respective tasks and that the project was very much a combined effort on the whole. We found that we worked very effectively as a team towards the end and certainly enjoyed collaborating on this task!

REFERENCES

[1] F. Mohr, "Using Python and Conditional Random Fields for Latin word segmentation," Medium.com. https://medium.com/@felixmohr/using-python-and-conditional-random-fields-for-latin-word-segmentation-416ca7a9e513 (accessed: May 30, 2021).

[2] H. Zhao, C.-N. Huang and M. Li, "An Improved Chinese Word Segmentation System with Conditional Random Field," in *Proc. 5th SIGHAN Workshop on Chinese Lang. Processing*, Sydney, Australia, Jul. 22-23, 2006, pp. 162-165.

[3] N. Xue, "Chinese Word Segmentation as Character Tagging," International Journal of Computational Linguistics & Chinese Language Processing, vol. 8, no. 1, pp. 29-48, Feb 2003.

[4] D. D. Palmer and M. A. Hearst, "Adaptive Multilingual Sentence Boundary Disambiguation," Computational Linguistics, vol. 23, no. 2, pp. 241-267, Jun. 1997.

[5] J. C. Reynar and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," in *Proc. 5th Conf. ANLP*, Washington, DC, USA, Mar. 31-Apr. 3, 1997, pp. 16-19, doi: 10.3115/974557.974561.

[6] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," Computational Linguistics, vol. 32, no. 4, pp. 485-525, Dec. 2006.

[7] D. Gillick, "Sentence Boundary Detection and the Problem with the U.S.," in *Proc. Human Language Technologies: 2009 Ann. Conf. NAACL*, Boulder, CO, USA, May 31-Jun. 5, 2009, pp. 241-244.

[8] D. J. Walker, D. E. Clements, M. Darwin and J. W. Amtrup, "Sentence Boundary Detection: A Comparison of Paradigms for Improving MT Quality," in *Proc. MT Summit VIII*, Santiago de Compostela, Spain, Sep. 18-22, 2001, pp. 18-22.

[9] J. Read, R. Dridan, S. Oepen, L. J. Solberg, "Sentence Boundary Detection: A Long Solved Problem?," in *Proc. COLING 2012: Posters*, Mumbai, India, Dec. 8-15, 2012, pp. 985-994.

[10] D. Griffis, C. Shiyade, E. Fosler-Lussier, A. M. Lai, "A Quantitative and Qualitative Evaluation of Sentence Boundary Detection for the Clinical Domain", in *Proc. AMIA Joint Summits Transl. Sci. 2016*, Jul. 2016, pp. 88-97.

[11] M. Stevenson and R. Gaizauskas, "Experiments on Sentence Boundary Detection," in *Proc. 6th Conf. ANLP*, Seattle, WA, USA, Apr. 29-May 4, 2000, pp. 84-89, doi: 10.3115/974147.974159.

[12] Y. Liu, A. Stolcke, E. Shriberg, M. Harper, "Using Conditional Random Fields for Sentence Boundary Detection in Speech," in *Proc. 43rd Ann. Conf. ACL (ACL '05)*, Ann Arbor, MI, USA, Jun 25-30, 2005, pp. 451-458, doi: 10.3115/1219840.1219896.

[13] P. Bedapudi, "DeepCorrection 1: Sentence Segmentation of Unpunctuated Text." Medium.com. 2018. https://praneethbedapudi.medium.com/deepcorrection-1-sentence-segmentation-of-unpunctuated-text-a1dbc0db4e98 (accessed May 26, 2021).

[14] B. Minixhofer, "NNSplit." Github.com. https://github.com/bminixhofer/nnsplit (accessed Jun. 8, 2021).

[15] O. Tilk and T. Alumäe, "Bidirectional Recurrent Neural Network with Attention Mechanism forPunctuation Restoration," in *Proc. Interspeech 2016*, San Francisco, CA, USA, Sep. 8-12, 2016, pp. 3047-3051, doi: 10.21437/Interspeech.2016-1517.

[16] W. Salloum, G. Finley, E. Edwards, M. Miller and D. Suendermann-Oeft, "Deep Learning for Punctuation Restoration in Medical Reports," in *Proc. 16th BioNLP*, Vancouver, Canada, Aug 4, 2017, pp. 159-164, doi: 10.18653/v1/W17-2319.

[17] T. Alam, A. Khan and F. Alam, "Punctuation Restoration using Transformer Models for High- and Low-Resource Languages," in *Proc. 6th W-NUT*, Online Conference, Nov. 19, 2020, pp. 132-142, doi: 10.18653/v1/2020.wnut-1.18.

[18] Ł.Augustyniak, P. Szymanski, M. Morzy, P. Zelasko, A. Szymczak, J. Mizgajski, Y. Carmiel and N. Dehak, "Punctuation Prediction in Spontaneous Conversations: Can We Mitigate ASR Errors with Retrofitted Word Embeddings?," in *Proc. Interspeech 2020*, Shanghai, China, Oct. 25-29, 2020, pp. 4906-4910, doi: 10.21437/Interspeech.2020-1250.

[19] A. Gravano, M. Jansche, M. Bacchiani, "Restoring Punctuation and Capitalization in Transcribed Speech," in *Proc. 2009 IEEE Conf.*

Acoustics, Speech and Signal Processing, Taipei, Taiwan, Apr. 19-24, 2009, pp. 4741-4744, doi: 10.1109/ICASSP.2009.4960690.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[21] D. Bahdanau, K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2016, *arXiv:1409.0473*.

[22] L.V. Lita, A. Ittycheriah, S. Roukos and N. Kambhatla, "tRuEcasIng," in *Proc. 41st Ann. Meet. ACL*, Sapporo, Japan, Jul. 7-12, 2003, pp. 152-159, doi: 10.3115/1075096.1075116.

[23] R. Grisham and B. Sundheim, "Message Understanding Conference-6," in *Proc. COLING 1996*, Copenhagen, Denmark, Aug. 5-9, 1996, pp. 466-471, doi: 10.3115/992628.992709.

[24] J. Huang and G. Zweig, "Maximum Entropy Model for Punctuation Annotation from Speech," in *Proc. Interspeech 2002*, Denver, CO, USA, Sep. 25-29, 2002, pp. 917-920.

[25] J.-H. Kim and P. C. Woodland, "Automatic capitalisation generation for speech input," in *Computer Speech & Language*, vol. 18, no. 1, pp. 67-90, Jan. 2004, doi: 10.1016/S0885-2308(03)00032-9.

[26] R. Rei, N. M. Guerreiro and F. Batista, "Automatic Truecasing of Video Subtitles Using BERT: A Multilingual Adaptable Approach," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, M. J. Lesot et al., Eds. Cham, Switzerland: Springer, 2020.

[27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, " BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL HLT 2019*, Minneapolis, MN, USA, Jun. 2-7, 2019, pp. 4171-4186.

[28] N. Reimers, "Truecaser." Github.com. https://github.com/nreimers/truecaser (accessed: Jun. 8, 2021).

[29] Stack Overflow, "How to split text without spaces into list of words," StackOverflow.com. https://stackoverflow.com/questions/8870261/how-to-split-text-without-spaces-into-list-of-words. (accessed: Jun. 2, 2021).

[30] S. Schweter and S. Ahmed, "Deep-EOS: General-Purpose Neural Networks for Sentence Boundary Detection," in *Proc. 15th Conf. NLP (KONVENS)*, Erlangen, Germany, Oct. 9-11, 2019, pp. 251-255.

[31] N. Reinerink, "BertPunc", GitHub.com. https://github.com/nkrnrnk/BertPunc. (accessed: May 30, 2021).

[32] O. Tilk, "Punctuator2." Github.com. https://github.com/ottokart/punctuator2 (accessed: Jun. 2, 2021).

[33] S. Bird, E. Loper and E. Klein, "nltk.tokenize package — NLTK 3.6.2 documentation," Nltk.org, 2021. https://www.nltk.org/api/nltk.tokenize.html (accessed Jun. 6, 2021).

[34] Stanford NLP Group, "stanfordnlp/stanza," GitHub.com, 2021. https://github.com/stanfordnlp/stanza. (accessed May 14, 2021).

[35] K. Papineni, S. Roukos, T. Ward and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in Proc. 40th Ann. Meet. ACL, Philadelphia, PA, Jul. 7-12, 2002, pp. 311-318, doi: 10.3115/1073083.1073135.

[36] Intento, "Intento API," GitHub.com. https://github.com/intento/intento-api. (accessed: Jun. 7, 2021).

ANNEX A

Early Output Example (Punctuation Restoration Stage Only)

The following is an example of early output obtained when testing the Punctuator component on its own. This output, along with early output for the Truecaser component, was primarily used as a basis for preliminary evaluation and discussion of our model in the presentation session that was given on 7th June 2021.

The text is taken from the following article from CBC: "Manitoba teachers will be able to get vaccine in U.S., premier says, but union says it shows there was no plan" (published 29th April 2021, available at: https://www.cbc.ca/news/canada/manitoba/manitoba-north-dakota-vaccination-teachers-1.6007257).

For the purposes of testing this component, the article text was preprocessed to remove headings, captions, etc., and prepared so that the text would be in the ideal output format envisaged after the space restoration stage (i.e. lowercase words separated by spaces without sentence delimiters, punctuation or capitalisation). This was then fed as input to the Punctuator component.

This output is discussed in further detail in the linguistic analysis subsection of the report. One feature of note here is that our model has output the restored punctuation marks directly in some cases and as tags surrounded by spaces in others (" .PERIOD ", " ,COMMA ". For the purposes of linguistic evaluation, these two different output formats were treated as identical.

Punctuator Output (2021-06-06):

manitoba teachers and other school employees will be able to get vaccinated in the us premier brian pallister says an agreement announced last week that allows manitoba truck drivers who regularly cross the border into north dakota to get vaccinated in the us will be extended to teachers and other school workers including janitors and administrators we want to make sure those schools are safe pallister said during a news conference thursday the person will go to the border cross get a vaccine and must come immediately back theyre not going shopping in grand forks pallister suggested a possible crossborder vaccination site for teachers could be near the emerson man border crossing another could be at the international peace garden that straddles the canada us border near boissevain in southwestern manitoba he said .PERIOD some of the details still need to be worked out a health canada spokesperson was unable to provide more information thursday one possibility is to have a vaccination hub open on weekends so teachers can make the trip without having to take time off pallister said when asked why the onus will be on teachers and education workers to drive across the border on their own time pallister said he thinks most teachers wont see it as a burden but an opportunity isolation requirements will be lifted for teachers and education system staff who opt to seek a vaccine this way pallister said currently anyone who enters manitoba must get tested and selfisolate for two weeks upon arrival federal rules require 14 days of quarantine for most canadians after they cross the border with some exemptions but the federal government is permitting exemptions for teacher vaccination he said the border has been closed to nonessential travel since early in the pandemic michelle driedger a community health sciences professor at the university of manitoba questioned why the initiative couldnt be managed as a popup ,COMMA clinic somewhere in manitoba .PERIOD she also said it and the travel exemption could send mixed messages i am sure that some teachers might respond to this announcement and be happy that there is an opportunity to go and get vaccinated she said. for many others it will be received as an insult to their profession and to their commitment to working with the province the manitoba government has been criticized in recent weeks for failing to include teachers on the priority list of workers who are eligible to be vaccinated the manitoba teachers society echoed those calls again thursday morning ahead of the announcement, in calling for all winnipeg schools to move to remote learning starting next week society president james bedford said the province hasn't consulted with the union on anything to do with vaccination he said the north dakota plan is really just an admission that there wasnt a plan to vaccinate those who work in our public school system bedford said the plan is only realistic for teachers who live within driving distance of the border have a vehicle and someone who can watch their own kids if they wanted to make the trip ,COMMA meanwhile cases in schools are rising .PERIOD there have been over 400 across 167 manitoba schools in the past two weeks about a quarter of them among staff manitoba's chief provincial health officer has consistently said contact tracing suggests most cases arent being acquired in schools but in the community winnipeg high school teacher lauren hope has her doubts she welcomes the chance for educators to get vaccinated but said making them travel isnt the answer shes also worried the change is coming too late to stop the transmission already happening in schools we know that the numbers reflect what happened weeks ago so if our numbers are high now and were really riding this third wave at a much higher level and rvalue than we did in october and november, then vaccines now arent enough hope said brandon university associate professor kelly saunders says the arrangement misses the mark. its yet another example of where this premier is just consistently tone deaf she said instead of prioritizing teachers and ensuring that they can get vaccinated he is now requiring them to travel across the border on their own time at their own expense saunders said .PERIOD resorting to a deal with north dakota fits into a broader narrative of how pallister has positioned the federal government as being responsible for bumps along the way in manitobas vaccine roll out theres a political angle here that pallister is trying to continually work and thats just part of the ongoing deflection and blameshifting cliff cullen manitobas education minister said in an ideal world the province would have enough vaccine doses from ottawa that it wouldnt need other partners i wish we had that vaccine in hand we dont so thats why were looking at various options he said after question period certainly north dakota has really stepped up and said they have some vaccine available asked why manitoba doesnt modify its eligibility requirements to vaccinate all teachers cullen said the province is relying on the advice of its vaccine task force which is now prioritizing all adults in covid 19 infection hotspots more details on the crossborder plan are expected next week .PERIOD

ANNEX B

Annotated Early Output Example (Punctuation Restoration Stage Only)

The following is a repeat of the output shown in Annex A that was obtained when testing the Punctuator component on its own, but with added annotation to indicate where the model correctly or incorrectly added a comma or period, or where these punctuation marks were missed and should have been added. This annotated output, along with early output for the Truecaser component, was primarily used as a basis for preliminary evaluation and discussion of our model in the presentation session that was given on 7th June 2021.

The text is taken from the following article from CBC: "Manitoba teachers will be able to get vaccine in U.S., premier says, but union says it shows there was no plan" (published 29th April 2021, available at: https://www.cbc.ca/news/canada/manitoba/manitoba-north-dakota-vaccination-teachers-1.6007257). When annotating the output for evaluation, the output was compared against the original text provided on this site.

For the purposes of testing this component, the article text was preprocessed to remove headings, captions, etc., and prepared so that the text would be in the ideal output format envisaged after the space restoration stage (i.e. lowercase words separated by spaces without sentence delimiters, punctuation or capitalisation). This was then fed as input to the Punctuator component.

The output has been manually annotated according to the following scheme:
#MISSEDAP#: Indicates a missed period that formed part of an abbreviation, e.g. "U.S.", "Man." (for "Manitoba")
#MISSEDEP#: Indicates a missed period that ended a sentence
#INCORRECTP#: Indicates a period that was inserted in an incorrect position
#CORRECTP#: Indicates a period that was inserted in a correct position
#MISSEDC#: Indicates a missed comma
#INCORRECTC#: Indicates a comma that was inserted in an incorrect position
#CORRECTC#: Indicates a comma that was inserted in a correct position

The following counts were obtained when running a Python script to count the number of tags inserted:
#MISSEDAP# = 7
#MISSEDEP# = 36
#INCORRECTP# = 2
#CORRECTP# = 5
#MISSEDC# = 35
#INCORRECTC# = 2
#CORRECTC# = 2

Punctuator Output (2021-06-06):

manitoba teachers and other school employees will be able to get vaccinated in the u#MISSEDAP#s#MISSEDAP##MISSEDC# premier brian pallister says#MISSEDEP# an agreement announced last week that allows manitoba truck drivers who regularly cross the border into north dakota to get vaccinated in the u#MISSEDAP#s#MISSEDAP# will be extended to teachers and other school workers#MISSEDC# including janitors and administrators#MISSEDEP# we want to make sure those schools are safe#MISSEDC# pallister said during a news conference thursday#MISSEDEP# the person will go to the border#MISSEDC# cross#MISSEDC# get a vaccine and must come immediately back#MISSEDEP# theyre not going shopping in grand forks#MISSEDEP# pallister suggested a possible crossborder vaccination site for teachers could be near the emerson#MISSEDC# man#MISSEDAP##MISSEDC# border crossing#MISSEDEP# another could be at the international peace garden that straddles the canada u#MISSEDAP#s#MISSEDAP# border near boissevain in southwestern manitoba#MISSEDEP# he said #INCORRECTP#.PERIOD some of the details still need to be worked out#MISSEDEP# a health canada spokesperson was unable to provide more information thursday#MISSEDEP# one possibility is to have a vaccination hub open on weekends so teachers can make the trip without having to take time off#MISSEDC# pallister said#MISSEDEP# when asked why the onus will be on teachers and education workers to drive across the border on their own time#MISSEDC# pallister said he thinks most teachers wont see it as a burden but an opportunity#MISSEDEP# isolation requirements will be lifted for teachers and education system staff who opt to seek a vaccine this way#MISSEDC# pallister said#MISSEDEP# currently#MISSEDC# anyone who enters manitoba must get tested and selfisolate for two weeks upon arrival#MISSEDEP# federal rules require 14 days of quarantine for most canadians after they cross the border#MISSEDC# with some exemptions#MISSEDC# but the federal government is permitting exemptions for teacher vaccination#MISSEDC# he said#MISSEDEP# the border has been closed to nonessential travel since early in the pandemic#MISSEDEP# michelle driedger#MISSEDC# a community health sciences professor at the university of manitoba#MISSEDC# questioned why the initiative couldnt be managed as a popup #INCORRECTC#,COMMA clinic somewhere in manitoba #CORRECTP#.PERIOD she also said it and the travel exemption could send mixed messages#MISSEDEP# i am sure that some teachers might respond to this announcement and be happy that there is an opportunity to go and get vaccinated#MISSEDC# she said#CORRECTP#. for many others it will be received as an insult to their profession and to their commitment to working with the province#MISSEDEP# the manitoba government has been criticized in recent weeks for failing to include teachers on the priority list of workers who are eligible to be vaccinated#MISSEDEP# the manitoba teachers society echoed those calls again thursday morning#MISSEDC# ahead of the announcement#CORRECTC#, in calling for all winnipeg schools to move to remote learning starting next week#MISSEDEP# society president james bedford said the province hasn't

consulted with the union on anything to do with vaccination#MISSEDEP# he said the north dakota plan is really just an admission that there wasnt a plan to vaccinate those who work in our public school system#MISSEDEP# bedford said the plan is only realistic for teachers who live within driving distance of the border#MISSEDC# have a vehicle and someone who can watch their own kids if they wanted to make the trip#MISSEDEP##INCORRECTC#,COMMA meanwhile#MISSEDC# cases in schools are rising #CORRECTP#.PERIOD there have been over 400 across 167 manitoba schools in the past two weeks#MISSEDC# about a quarter of them among staff#MISSEDEP# manitoba's chief provincial health officer has consistently said contact tracing suggests most cases arent being acquired in schools but in the community#MISSEDEP# winnipeg high school teacher lauren hope has her doubts#MISSEDEP# she welcomes the chance for educators to get vaccinated#MISSEDC# but said making them travel isnt the answer#MISSEDEP# shes also worried the change is coming too late to stop the transmission already happening in schools#MISSEDEP# we know that the numbers reflect what happened weeks ago#MISSEDC# so if our numbers are high now and were really riding this third wave at a much higher level and rvalue than we did in october and november#CORRECTC#, then vaccines now arent enough#MISSEDC# hope said#MISSEDEP# brandon university associate professor kelly saunders says the arrangement misses the mark#CORRECTP#. its yet another example of where this premier is just consistently tone deaf#MISSEDC# she said#MISSEDEP# instead of prioritizing teachers and ensuring that they can get vaccinated#MISSEDC# he is now requiring them to travel across the border on their own time at their own expense#MISSEDEP# saunders said #INCORRECTP#.PERIOD resorting to a deal with north dakota fits into a broader narrative of how pallister has positioned the federal government as being responsible for bumps along the way in manitobas vaccine roll out#MISSEDEP# theres a political angle here that pallister is trying to continually work and thats just part of the ongoing deflection and blameshifting#MISSEDEP# cliff cullen#MISSEDC# manitobas education minister#MISSEDC# said in an ideal world#MISSEDC# the province would have enough vaccine doses from ottawa that it wouldnt need other partners#MISSEDEP# i wish we had that vaccine in hand#MISSEDC# we dont#MISSEDC# so thats why were looking at various options#MISSEDC# he said after question period#MISSEDEP# certainly#MISSEDC# north dakota has really stepped up and said they have some vaccine available#MISSEDEP# asked why manitoba doesnt modify its eligibility requirements to vaccinate all teachers#MISSEDC# cullen said the province is relying on the advice of its vaccine task force#MISSEDC# which is now prioritizing all adults in covid 19 infection hotspots#MISSEDEP# more details on the crossborder plan are expected next week #CORRECTP#.PERIOD

ANNEX C

Early Output Example (Truecaser Stage Only)

The following is an example of early output obtained when testing the Truecaser component on its own. This output, along with early output for the Punctuator component, was primarily used as a basis for preliminary evaluation and discussion of our model in the presentation session that was given on 7th June 2021.

The text is taken from the following article from CBC: "Manitoba teachers will be able to get vaccine in U.S., premier says, but union says it shows there was no plan" (published 29th April 2021, available at: https://www.cbc.ca/news/canada/manitoba/manitoba-north-dakota-vaccination-teachers-1.6007257).

This output is discussed in further detail in the linguistic analysis subsection of the report. For the purposes of testing this component, the article text was preprocessed to remove headings, captions, etc., and prepared so that the text would be in the ideal output format envisaged after the punctuation stage component (i.e. lowercase words separated by spaces with commas and periods restored, but without delineated sentence boundaries or capitalisation). This was then fed as input to the sentence splitter component, which was in turn fed to the Truecaser component.

Note that since the sentence splitter component used at the time of testing was a simple rule-based solution, it has also identified abbreviations with periods such as "U.S." not immediately followed by a comma as sentence-ending punctuation and erroneously considered these to be followed by the start of a new sentence. A preprocessing error also meant that a space was missed between the sentences "…in Southwestern Manitoba. He said…", meaning that the sentence did not split as it should have done here. A line split also did not occur as expected at "…immediately back. Theyre…".

Sentence boundaries are indicated by the start of a new line, which has been annotated here with the characters "\n" for visual clarity.

Truecaser Output (2021-06-07):

Manitoba teachers and other school employees will be able to get Vaccinated in the U.S., Premier Brian Pallister says.\n
An agreement announced last week that allows Manitoba truck drivers who regularly cross the border into North Dakota to get Vaccinated in the U.S.\n
Will be extended to teachers and other school workers, including janitors and administrators.\n
We want to make sure those schools are safe, Pallister said during a news conference Thursday.\n
The person will go to the border, cross, get a vaccine and must come immediately back. Theyre not going shopping in Grand Forks.\n
Pallister suggested a possible Cross-Border vaccination site for teachers could be near the Emerson, Man., Border crossing.\n
Another could be at the international peace garden that Straddles the Canada-U.S.\n
Border near Boissevain in Southwestern Manitoba.He said some of the details still need to be worked out.\n
A health Canada Spokesperson was unable to provide more information Thursday.\n
One possibility is to have a vaccination Hub open on weekends so teachers can make the trip without having to take time off , Pallister said.\n
When asked why the onus will be on teachers and education workers to drive across the border on their own time, Pallister said he thinks most teachers wont see it as a burden but an opportunity.\n
Isolation requirements will be lifted for teachers and education system staff who Opt to seek a vaccine this way, Pallister said.\n
Currently, anyone who enters Manitoba must get tested and Self-Isolate for two weeks upon arrival.\n
Federal rules require 14 days of Quarantine for most Canadians after they cross the border, with some exemptions, but the Federal government is permitting exemptions for teacher vaccination, he said.\n
The border has been closed to Non-Essential travel since early in the pandemic.\n
Michelle Driedger, a community health sciences professor at the University of Manitoba, questioned why the initiative Couldnt be managed as a Pop-Up clinic somewhere in Manitoba.\n
She also said it and the travel exemption could send mixed messages.\n
I am sure that some teachers might respond to this announcement and be happy that there is an opportunity to go and get Vaccinated, she said.\n
For many others it will be received as an insult to their profession and to their commitment to working with the province.\n
The Manitoba government has been criticized in recent weeks for failing to include teachers on the priority list of workers who are eligible to be Vaccinated.\n
The Manitoba teachers society echoed those calls again Thursday morning, ahead of the announcement, in calling for all Winnipeg schools to move to remote learning starting next week.\n
Society President James Bedford said the province Hasnt consulted with the Union on anything to do with vaccination.\n
He said the North Dakota plan is really just an admission that there Wasnt a plan to Vaccinate those who work in our public school system.\n
Bedford said the plan is only realistic for teachers who live within driving distance of the border, have a vehicle and someone who can watch their own kids if they wanted to make the trip.\n
Meanwhile, cases in schools are rising.\n
There have been over 400 across 167 Manitoba schools in the past two weeks, about a quarter of them among staff.\n

Manitobas chief provincial health officer has consistently said contact tracing suggests most cases Arent being acquired in schools but in the community.\n
Winnipeg high school teacher Lauren hope has her doubts.\n
She welcomes the chance for educators to get Vaccinated, but said making them travel Isnt the answer.\n
Shes also worried the change is coming too late to stop the transmission already happening in schools.\n
Asked why Manitoba Doesnt modify its eligibility requirements to Vaccinate all teachers, Cullen said the province is relying on the advice of its vaccine task force, which is now Prioritizing all adults in Covid-19 infection Hotspots.\n
More details on the Cross-Border plan are expected next week.\n