Identifying Robust Markers of Parkinson's Disease in Typing Behaviour Using a CNN-LSTM Network

Neil Dhir[1][2][†], Mathias Edman[1][†], Álvaro Sánchez Ferro[3], Tom Stafford[4], Colin Bannard[5]
[1]Kamin AI, [2]Alan Turing Institute, [3]HM CINAC,
[4]University of Sheffield, [5]University of Liverpool
{neil,mathias}@kamin.ai, alvarosferro@hotmail.com
t.stafford@sheffield.ac.uk, cbannard@liverpool.ac.uk

Abstract

There is urgent need for non-intrusive tests that can detect early signs of Parkinson's disease (PD), a debilitating neurodegenerative disorder that affects motor control. Recent promising research has focused on disease markers evident in the fine-motor behaviour of typing. Most work to date has focused solely on the timing of keypresses without reference to the linguistic content. In this paper we argue that the identity of the key combinations being produced should impact how they are handled by people with PD, and provide evidence that natural language processing methods can thus be of help in identifying signs of disease. We test the performance of a bi-directional LSTM with convolutional features in distinguishing people with PD from age-matched controls typing in English and Spanish, both in clinics and online.[1]

1 Introduction

Parkinson's disease is a neurodegenerative disease that affects approximately 1% of people over the age of 60 (De Lau and Breteler, 2006). Its cardinal manifestations include bradykinesia (slowness of movement), tremor and rigidity. These result from the degeneration of dopaminergic neurons in the basal ganglia (an area of the brain responsible for action selection). A particular challenge in the treatment of PD is that by the time such motor signs are present, over 50% of neurons in the affected area of the basal ganglia (the substantia nigra) have been lost (Fearnley and Lees, 1991). While neuroimaging can pick up on these changes (Barber et al., 2017), such procedures are prohibitively expensive and cannot be performed on whole populations. There is thus an urgent need for cheap and easy-to-administer measures that can be used for the identification of at-risk individuals.

A long-used simple motor test for PD is the alternating finger tapping test (Burns and DeJong, 1960). This test involves asking a person to alternately tap an index finger in two locations a set distance apart on a surface or on a keyboard (Giovannoni et al., 1999). People with PD are typically able to perform fewer taps over a 30 second period than people with no diagnosis. While such measures have proved useful, they suffer from a clear lack of specificity – slowing of movement is also a strong predictor of other neurodegenerative disorders, such as Alzeimer's disease (Roalf et al., 2018). Furthermore, neural degeneration is unlikely to be detected by as coarse-grained a measure as tapping rate until the disease is relatively advanced. If specificity and earlier detection is to be achieved more targeted tests will be required.

There is good theoretical reason to think that more PD-specific markers will be present in recordings of learned serial order behaviours, such as making a cup of tea, driving or typing. Analysis of the production of such frequently-performed behaviours, and their underlying neurobiology, often distinguishes between habit (the automatic production of routinised movements) and goal-directed responses (behaviours that involve top-down planning; Dolan and Dayan 2013). There is substantial evidence that the degeneration of the basal ganglia in PD primarily affects areas responsible for automatic behaviours (Sharman et al., 2013), and results in a shift in the balance of habitual and goal-directed control (Hadj-Bouziane et al., 2013). Redgrave et al. (2010) predict that people with early-stage or prodromal PD will have a problem initiating their automatic behaviours.

This paper focuses on the detection of markers of PD in one such behaviour – that of typing. It is motivated by the prediction that people with PD

[†]Equal contribution.

[1]Code, models and data used in this paper can be found at: http://typingresearch.com/conll2020/

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 578–595
Online, November 19-20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

will, from very early on and potentially prodromally (before the emergence of the acute symptoms that allow conventional diagnosis), change the way that they type, as they lose capacity for automatic control. Natural language processing provides us with techniques that we can use to pick up on those changes. As a motor behaviour, typing has been the focus of previous work on detecting or monitoring PD (see related work section). Some such work has focused on coarse-grained measures such as typing speed. These suffer from the lack of specificity associated with tapping measures. Other promising work has looked at more detailed timing measures. However this has continued to ignore the identity of the sequences being typed. Different sequences of keys present different motor challenges due to the position of the keys on the keyboard and the hand used. The extent to which typing these will be facilitated by prior automatisation depends on the relative frequency with which they have been typed (Behmer and Crump, 2016). We therefore expect consideration of the content of typing to be critical in picking up on PD-related changes.

We describe a method for using a convolutional neural network (CNN) long short term memory (LSTM) network to distinguish people with PD from age-matched people with no diagnosis. One motivation for this choice is that we want to pick up on the fine temporal details of the sequential data, in order to provide a measure that will be specific to PD. However the difficulty in picking up on such subtle information is that any typing dataset will also contain cruder information, such as the average differences in overall timing across participants. In order to tackle this, we normalise all temporal variables using robust scaling – subtracting each participant's median value from all their datapoints and dividing by their interquartile range. We thereby require our network to pick up on more subtle and potentially disease-specific information.

The contributions of this article are as follows:

- We introduce a new task and data type of urgent clinical importance.
- We show that when we remove coarse-grained differences between people with PD and controls we are able to detect a strong (and, we suggest, more disease-specific) signal.
- We provide evidence that adding character information to a CNN-LSTM that contains only timing information improves performance across datasets.

2 Related work

A simple motor test that we might consider a precursor to the use of typing is the alternating finger tapping test (Burns and DeJong, 1960). This test involves asking a person to alternately tap an index finger in two locations a set distance apart on a surface or on a keyboard (Giovannoni et al., 1999). Noyce et al. (2014) report that the number of key taps in 30 seconds (averaged across hands) can be used to distinguish patients from controls, identifying 50% of true positives with only 15% false positives. Using the same measure (selecting the worst performing limb in patients and comparing it with the best performing limb in controls), Hasan et al. (2019) report an AUC of 0.87.

While tapping tests are widely used they suffer from a lack of specificity to PD. While they distinguish people with PD from control participants with considerable success, there is good reason to think that they will struggle to distinguish PD from other neurodegenerative disorders. Roalf et al. (2018) report an AUC of 0.68 in distinguishing people with PD from people with Alzheimer's disease using a single tapping test.

In pursuit of an easier-to-gather alternative to finger-tapping tests, Austin et al. (2011) examine the interkey intervals (IKIs) of people typing usernames while logging-in to a website. They found a moderate-to-strong correlation between the participants' median IKIs during typing and the mean time between taps finger taps in a 10 second period. Building on this, Giancardo et al. (2016) used key-hold times during transcription typing in order to distinguish 42 people with recently-diagnosed PD (off medication) from 43 controls. Properties of the distribution of hold times for each patient was used in an ϵ-support regression to generate a unique score. In a two-fold cross-validation this achieved a combined AUC of 0.81, comparable to an AUC of 0.75 achieved with an alternating finger tapping test on the same sample. Adams (2017) logged key events during regular computer use over an extended period by 20 patients and 33 controls. Information about hold times and IKIs, including measures of variance and of asymmetry between hands was used in a classification ensemble of eight different classification methods. This ensemble, trained on the new data, achieved an AUC of 0.97 on the 85 participants from Giancardo et al. (2016).

All of the work described above has represented typing behaviour with summary statistics rather than as sequences. Furthermore they have analysed the timing of keystrokes without considering what is being typed. The one exception to this latter point is the work of Bannard et al. (2019) who look at the accuracy of typing while copying text using engineered features. They predict that people with PD, while making more errors in general, should make fewer 'habit slips'. This is when a well-learned sequence of key presses is produced in an inappropriate context, such as typing `t-h-i-n-g` when the intended word is `t-h-i-n` because `-i-n-g` is a frequent sequence. They find that is the case, and that adding this information to a generalised additive regression model predicting disease progression gives an improvement in fit relative to a model just including timing information.

3 Datasets

We perform analyses of the following three datasets, representing two different usage contexts (recruited and tested in a clinic, and recruited and tested remotely online) and two different languages: English and Spanish. All participants were tested via a browser-based app which presents a series of sentences to be copy-typed, and collects the identity and timing of each key-press. All datasets contain information about key down timing (when the typist pressed each key), and the online-recruited dataset additionally contains information about about key up timing (when they released each key). Summary statistics are found in table 1.

Table 1: Summary statistics of the datasets under investigation. Here N_p^+ refers to the number of PD patients (unmedicated and medicated) with N_p^- referring to the number of control participants. N_s^+ is the number of sentences typed by PD positive patients.

Collection	Language	N_p^+/N_p^-	N_s^+/N_s^-
In-clinic	English	16/25	426/739
In-clinic	Spanish	11/9	310/265
Online	English	99/130	1415/1862

3.1 In-clinic English copy-typing

Sixteen patients and 25 age-matched controls were recruited and tested during a visit to a hospital clinic in the UK (see Bannard et al. 2019). Patients were recruited to be in the early stages of PD (Hoehn-Yahr stages $0 - 2.5$, UPDRS < 20 in the medicated state), with normal cognitive function and < 5 years from a confirmed diagnosis. All patients were asked to type 15 sentences, all of which were taken from English-language Wikipedia articles, and ranged from 10 to 25 words (average of $\mu = 19$ words) in length. The experimental protocol was approved by NHS Health Research Authority (no. STH18662TK). All participants were tested twice – once before taking their morning medication and once after for patients. On a five point self assessment of their typing ability, ranging from none to secretarial proficiency, control participants reported an average 3.1 (4% no experience) and patients reported an average 2.7 (12% no experience).

3.2 In-clinic Spanish copy-typing

Eleven patients and nine age-matched controls were tested during a visit to a hospital clinic in Spain (see Bannard et al. 2019). The inclusion criteria for patients was the same as for the clinic-tested English sample. All patients were asked to type 30 sentences, all of which were taken from Spanish language Wikipedia articles, and ranged from 12 to 25 words (average of $\mu = 18$ words) in length. The experimental protocol was approved by HM Hospitales, Spain (no. 14.11.710-GHM). Participants were tested only once. Six of the patients were tested prior to taking their morning medication and five after. On a five point self assessment of their typing ability, ranging from none to secretarial proficiency, control participants reported an average 3.9 (0 had no experience) and patients reported an average 3.7 (0 had no experience).

3.3 Online English copy-typing

For this newly-collected dataset, 130 controls and 100 people with PD were recruited and tested online. The people with PD were recruited via the recruitment service of a major US-based Parkinson's charity. The control participants were recruited via a participant recruitment service. All participants were aged between 50 and 90 and identified as resident in the US. Patients were recruited to be self-reportedly in the early stages of PD (Hoehn-Yahr stages $0 - 3$ as indicated by responses to a questionnaire), and within five years of a diagnosis. The sentences typed were the same as those typed by the in-clinic English sample. The experimental protocol was approved by the University of Liverpool Ethics Committee (no. 4572). Of the 100 people with PD, 24 reported that they either do not take medication or had not taken any medication yet that day. On a five point self assessment of their typing ability, ranging from novice to expert, control participants reported an average 2.6 (13% novices), medicated people with PD an average 3 (4% novices), and unmedicated people with PD an average 3.1 (8% novices). Note that in contrast to what we see in the clinic-collected datasets, the people with PD here rate their typing ability more highly than the controls. Unlike the in-clinic samples, this dataset contains information about both key down timing and key up timing.

4 Method

We implement a neural language model (NLM) which receives two different types of information in variety of combinations: (1) *Character identity information*: one-hot encoded character sequences; continuous bag-of-words model (Mikolov et al., 2013) encoded character sequences; (2) *Keypress timing information*: inter-key interval (IKI), time elapsed between consecutive key down events; hold-time, time elapsed between key down and key up events for a specific character; pause, time difference between key up and key down events for consecutive key presses. The temporal information is shown pictorially in fig. 1.

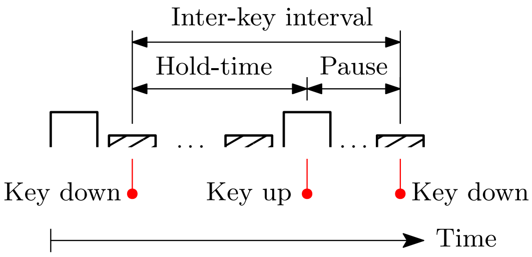


Figure 1: Pictorial description of the compression and release of keyboard keys and the temporal information that results from those actions.

Different timing information is available in different datasets as reported in §3. We adapt our data representation accordingly. We are interested here in the value of character information over timing, and examine its utility by building models with just timing and then with timing and character.

Our approach is inspired by the recent work of Kim (2014); Kim et al. (2016); Zhang et al. (2015). The main component is the temporal convolutional module (Zhang et al., 2015), which computes a one-dimensional (1D) convolution over characters. Convolutional neural networks (CNN) employ layers with convolving filters (Kim, 2014) which are applied to local features (derived in our case from the above information list of textual information). Diverging from their approach, we use a smaller number of convolutional layers followed by a bidirectional long short-term memory (LSTM) layer (Schuster and Paliwal, 1997). As such, our architecture is able to extract both local and global features as described in the work by Zhou et al. (2015) who utilise a similar architecture. For a detailed description of the model architecture see fig. 2 and appendix B.

4.1 Data representation

Our model takes sentences (as sequence of characters and/or key press timing information) as input. Before introducing the construction process of sentence sequences we shall give a detailed description of its elements.

First, for the sake of comparison, we conduct experiments with two different character-identity representations. The default representation is one-hot encoding of characters where each unique character is associated with an index i such that the representation of a character is a binary vector $\mathbf{c}$ where $c_i = 1$ and $c_j = 0$, $\forall j \neq i$. We also evaluate using a continuous vector representation of characters, which is an adaptation of the commonly-used continuous bag-of-words (CBOW) embedding (Mikolov et al., 2013). While for word embeddings the CBOW algorithm learns the representation by predicting words from the surrounding context, our character level adaptation utilises the same algorithm but for the task of predicting characters from their context. We learn the character embeddings from a corpus of 100,000 Wikipedia articles[2], such that we obtain a character dictionary where each character is associated with a unique continuous vector representation of 50 dimensions.

The datasets in §3 contain, in addition to the characters used, a timestamp for each character key-down press t^d. However only for the online English copy-typing dataset in §3.3 are timestamps for key-up events denoted t^{up}, available. We define the order of a character sequence as the order of the associated key-down timestamps indexed by k such that $t^d_{k-1} \leq t^d_k$, $\forall k \in 1, \ldots, K$. For most end-to-end deep learning one typically omits feature engineering and let the networks learn feature representations from large datasets. Here however we are dealing with relatively small (in the context of deep learning) datasets. In particular the

[2] https://blog.lateral.io/2015/06/the-unknown-perils-of-mining-wikipedia/

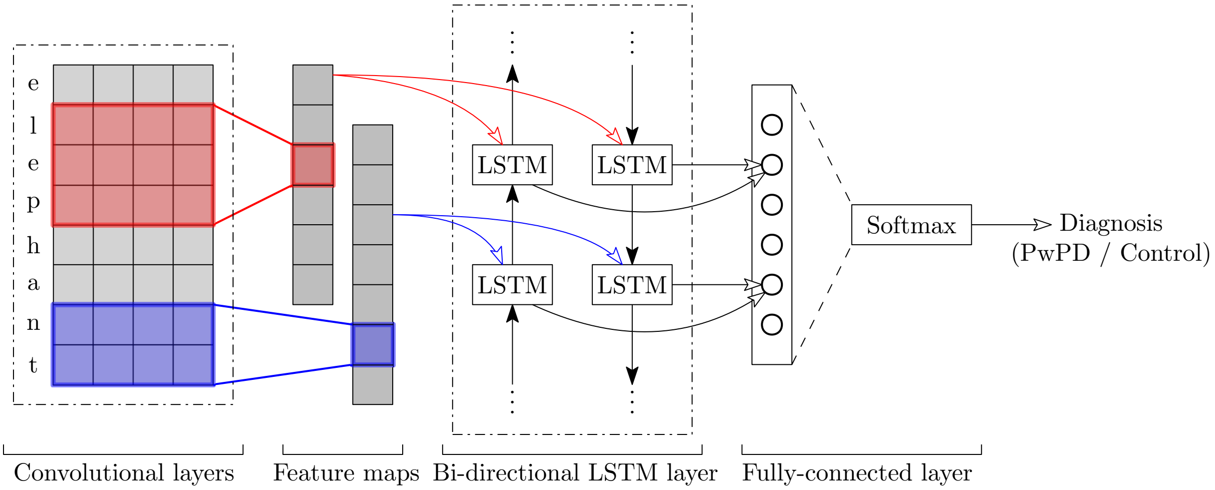


Figure 2: Architecture of the neural language model applied to the example word: `elephant`. On the left is shown the encoding matrix $\mathbf{X}^n$ where each row, as shown, corresponds to a character. From left to right, we see the elements of the NLM including convolutional layers, a bi-directional LSTM layer and finally a fully connected output layer with Softmax activation.

In-clinic English and Spanish datasets contain just 1165 and 575 sentences respectively (see table 1 for further details). To aid learning we thus engineer a set of three features from the key press timestamps.

As discussed in section §1 and §2 we expect the effects of bradykinesia among people with PD (PwPD) to result in differences in the average timings of keystrokes. However the goal of our experiments is not to maximise performance on any one dataset, but rather to find evidence of more robust PD-specific typing characteristics that can help improve the specificity of PD detection systems. To this end we attempt to mute the coarse-grained, between-group differences in our data by employing participant-level standardisation of all timing related features. This is done by computing the median and interquartile range of all key press timing features for each participant, and then robustly scaling their corresponding sentences by subtracting the median and dividing by the interquartile range.

Finally a sentence is represented as

$$\mathbf{x}^n_{1:K} = x_1 \oplus x_2 \oplus \cdots \oplus x_k \oplus \cdots \oplus x_K \qquad (1)$$

where $\oplus$ is the concatenation operator, where n indexes each sentence, k indexes each character within each sentence and x is the character identity encoding vector (one-hot or CBOW) appended with the timing features associated with the key press of that character keypress. The longest sentence in any dataset has length K_{max}, and any encoded sentence $K_n < K_{\text{max}}, \forall n \in \{1, \ldots, N\}$ is padded with $|K_{\text{max}} - K_n|$ all-zero vectors so that all encoded sentences $\mathbf{X}^n$, have the same size: $\mathbf{X}^n \in [0, 1]^{K_{\text{max}} \times m}$.

4.2 Text pre-processing

Here we will briefly discuss the most important preprocessing steps. The complete procedure, with detailed description, can be found in appendix A. First, following the recommendation of Zhang et al. (2015), all sentences are converted to lower-case. Second, in this study we partially 'implement' the error correction employed by the participant. While we are interested in the errors that participants make, and indeed Bannard et al. (2019) show that the error types made can be indicative of disease status, we assume that the process by which they notice and correct those errors will be idiosyncratic and not informative regarding our classification goals. Consider the following example sentence, taken from the dataset described in §3.1:

> Books include Penguin
> Island, a satire on the
> F✗Dreyfus afffair✗✗✗✗air.

Here the user has employed five corrective actions (backspaces) which we indicate with ✗. For each sentence we implement and then delete all but one of these backspace actions leaving only the first errorfully pressed key (the first `f` in the `fff`) and a single backspace symbol. Thus the text becomes:

> Books include Penguin
> Island, a satire on the
> F✗Dreyfus aff**f**✗air.

The single correction character ✗ is left in the text to be used as indicators for the NLM in the downstream classification task. When only a single corrective action occurs it is simply left unamended as shown above. For an example see fig. 5 where the correction character passed to the NLM is ω.

5 Experimental setup

The purpose of our experiments is to understand the effect of including character information in the classification of PD patients, when employing copy-typing as a diagnosis medium. Using the model discussed in §4 we conduct multiple binary-classification experiments to distinguish sentences written by people with PD (PwPD), from those written by age-matched controls. The same exercise is undertaken to classify participants themselves.

We evaluate performance by measuring the area under the receiver operating characteristic curve (AUC). This is a common approach when dealing with a two-class prediction problem (binary classification), in which the outcomes are labelled either as positive (PwPD) or negative (control). The AUC scores reported in §6 are calculated on the test sets. For sentence classification we use participant level five-fold cross-validation ensuring that sentences from any one participant do not exist in both the train and test set. We report the mean and standard deviation over folds. For participant classification we aggregate the sentence classification probabilities using logistic regression with leave-one-out cross validation and employ bootstrapping to report mean and standard deviation. The model is applied to the datasets described in detail in §3 with summary statistics given in table 1.

We conduct hyperparameter search, model introspection and ablation studies. Each dataset is preprocessed according to the procedures outlined in §4.1 and §4.2, and split into train, test and validation sets. This partitioning reduces the number of samples which can be used for learning the model. Our datasets are small compared to those typically used for deep learning. We deal with this in multiple ways, as detailed in appendix C.

6 Results

Our main experiments, as outlined above, involve the use of timing information that has been robustly scaled at the participant level in order to remove coarse-grained differences between groups. To aid understanding of the data, however, we will first report the performance of a classifier that uses the information that we have removed - the median and interquartile range of keypresses - as the sole features. The AUCs for logistic regressions using these features as predictors can be seen in table 3.

Table 3: Results from logistic regression models with median and interquartile range for interkey intervals as features. We report mean AUC (and SDs) for both medicated PwPD vs. controls in the On columns and unmedicated PwPD vs. controls in the Off column. The Spanish PwPD are mixed in medication status but treated as a single group due to the small sample size.

	Sentence classification	
Dataset	Off	On
In-clinic English	0.76 (0.14)	0.76 (0.11)
Online English	0.64 (0.11)	0.53 (0.04)
In-clinic Spanish	0.91 (0.13)	N/A
	Participant classification	
Dataset	Off	On
In-clinic English	0.77 (0.08)	0.76 (0.08)
Online English	0.56 (0.07)	0.56 (0.04)
In-clinic Spanish	0.91 (0.09)	N/A

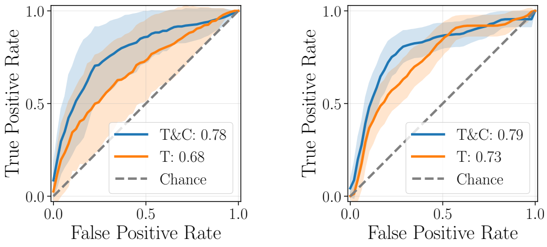
As can be seen the performance of these classifiers is good in some cases, particularly for the in-clinic Spanish dataset. However performance is variable, being poorest for the online dataset. This pattern of results is to be expected and our goal here is not to surpass their performance but to see how we can perform with more PD-specific features. The results for our main models, using the robust-scaled data, are reported in table 2 and fig. 3. For all datasets, the addition of character information gives an improvement in performance over timing-only models. The dataset on which the simple IKI summary-statistic models reported above do worst (the online English dataset) is the dataset on which the best performance is reported here. This is likely because it is the largest dataset and thus the best suited to deep learning methods. This suggests that performance improvements will be possible for the network models with larger datasets.

6.1 Model interpretation

Deep learning models are often criticised for being black box machines and the interpretation of deep learning techniques is a growing area of interest (Buhrmester et al., 2019). We apply one such technique – Gradient-weighted Class Activation Mapping (Grad-CAM; Selvaraju et al. 2017) – to our model. Grad-CAM is commonly used to analyse how CNN-based computer vision models make decisions and highlight regions in the image that the

Table 2: Results from NLM experiments showing improved performance for inclusion of character identity information across all datasets. We report mean AUC (and SDs) for both medicated PwPD vs. controls in the *On* columns and unmedicated PwPD vs. controls in the *Off* column. The Spanish PwPD are mixed in medication status but treated as a single group due to the small sample size.

	Sentence classification		Participant classification	
Dataset	Off	On	Off	On
In-clinic English				
Time Only	0.56 (0.09)	0.59 (0.07)	0.47 (0.09)	0.58 (0.09)
Time and Character (one-hot)	**0.64 (0.03)**	**0.66 (0.07)**	**0.65 (0.09)**	**0.80 (0.07)**
Time and Character (CBOW)	0.62 (0.05)	0.65 (0.11)	0.64 (0.09)	0.70 (0.08)
Online English				
Time Only	0.68 (0.16)	0.64 (0.10)	0.73 (0.06)	0.65 (0.04)
Time and Character (one-hot)	**0.78 (0.14)**	**0.70 (0.04)**	0.79 (0.06)	0.70 (0.04)
Time and Character (CBOW)	0.77 (0.13)	0.67 (0.08)	**0.84 (0.05)**	**0.75 (0.04)**
In-clinic Spanish				
Time Only	0.51 (0.13)	N/A	0.68 (0.13)	N/A
Time and Character (one-hot)	**0.68 (0.11)**	N/A	**0.77 (0.12)**	N/A



(a) MRC Sentence

(b) MRC Participant

Figure 3: ROC curves for sentence-level (a) and participant (b) classification for the (one-hot) online English *Off*-medication datasets are included.

model deems important. We repurpose Grad-CAM to analyse which part of a sentence a CNN-based NLM deems important for classification. For illustration we have included an example visualisation where we have applied our model to a sentiment analysis task where Grad-CAM highlights the parts of the sentence that indicate it should be classified as having positive sentiment – see fig. 4. We use the same approach to produce visualisations that highlight the parts of a sentence that our model uses to make a distinction between PwPD and Controls.

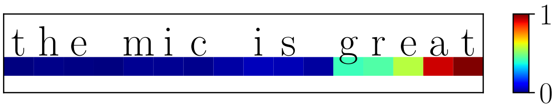


Figure 4: Example of Grad-CAM visualisation where we apply our network to a sentiment analysis task. Here we see the Grad-CAM highlighting the word "great" as important for determining that the sentence has positive sentiment.

Grad-CAM plots for example PwPD and control participants from the online English dataset for each of our sentences can be found in appendix D. These images show the gradients of the second and final convolutional layer for the positive diagnosis ("typist has PD") classification class. Grad-CAM visualisations for all participants and sentences, all convolutional layers and all classification classes, can be accessed at `http://typingresearch.com/conll2020/`. An illustrative two-word excerpt from one of our sentences typed by a single PwPD can be seen in figure fig. 5. The first word `different` (typed with a single corrected error on the second character by this typist) is mostly blue indicating that there is little in the sequence of keystrokes that the model takes as indicating that it was typed by a PwPD, while the second word `pronunciation` spans more colours indicating that it contains keystrokes that are indicative of its being typed by a PwPD.

Looking across participants we see that certain parts of the sentences are consistently more important for classification than others, as indicated by their having gradients that diverge between PwPD and controls. The first thing that this illustrates is that key identity matters, confirming the conclusions of our ablation study. It also allows us to look at what properties the most discriminative key sequences have in common. While no single property can be identified as the clearest marker of PD, we can identify suggestive patterns that are useful in understanding typing in this population. This

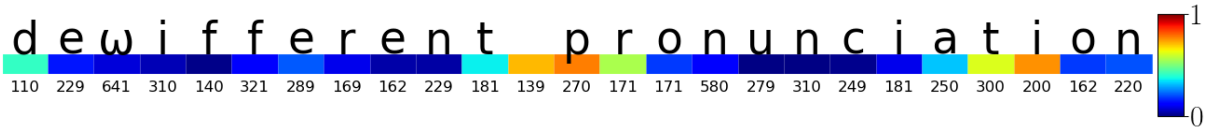


Figure 5: Grad-CAM visualisation of our network applied to a sentence typed by one participant, here ω stands in for a correction to the text. The colours represent the importance of the different parts of the string for determining that the typist has PD. The bottom numbers are the inter-key intervals between typed characters.

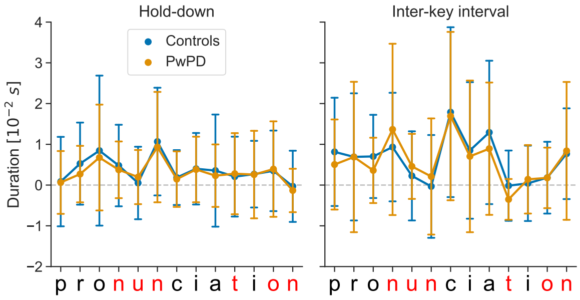


Figure 6: Plot showing the mean and standard deviations for the participant-scaled key hold and inter-key intervals for each character in the word. The black (left) and red (right) colouring indicates which hand typed that character.

exploration serves as an illustration of how model interpretation can provide potential mechanistic hypotheses to be tested in future work.

One pattern that the network seems to pick up on can be seen in fig. 5 by observing the changes in gradients for the keys with respect to the interkey intervals seen below the sequence. There is a sequence of keys with high gradients at the end of the first word and beginning of the second that have relatively low IKIs. A notable property of this subsequence is that each of the keys is typically typed with a different hand from the previous key. Figure 6 shows the mean and standard deviations for (robustly-scaled) key hold times and inter-key intervals for the word `pronunciation`. The letters are colour coded according to whether the character is typed with the same hand (red) as the preceding character or the other hand (black). This is based on the approximation that the leftmost 5 columns of the keyboard (from Q, A and Z to T, G and B) are typed with the left hand and the rightmost 5 columns are typed with the right (Feit et al., 2016). Moving between keys when switching hands is fairly straightforward to perform while switching between keys with the same hand requires considerable agility. We observe in our data that the typing speed of PwPD is differentially affected by this more than that of the controls. The network picks up on this and has a tendency toward higher gradients at between-hand transitions where the typing speed has a relative dip for a participant. Figure 7 provides further illustration of this widespread pattern.

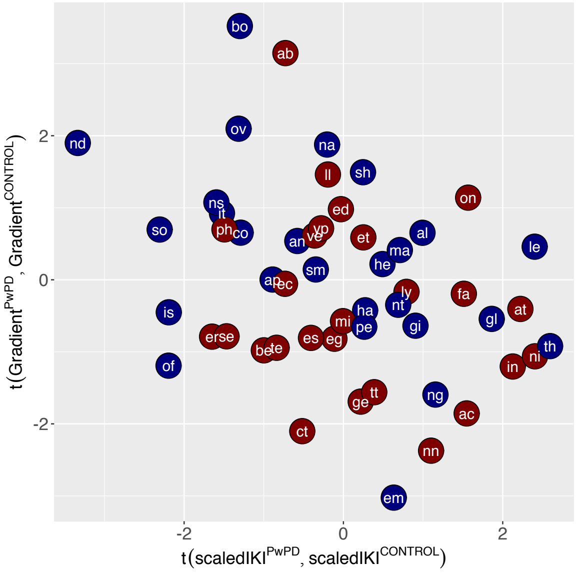


Figure 7: Key transitions for a single example sentence plotted along three dimensions. The $y-$axis indicates the discriminative ability of each key transition, indicated by the t-value for a comparison of the gradients for that keypress in patients and controls, such that a high value indicates that patients have consistently higher gradients than controls. The $x-$axis represents the extent to which the distribution of scaled IKIs for each keystroke are differ between patients and controls, again using a $t-$test (so that a high value indicates that patients have more consistently higher scaled IKIs than controls). The circles containing bigrams that involve a within-hand transition are shown in red and the circles containing across-hand-transition bigrams are shown in blue. The bigrams that have highest values on the y scale (that have gradients that are most consistently higher in patients in controls) are those that have a lower a value on the $x-$scale (they are associated with a relative dip in IKI in patients that is consistently more pronounced than anything seen in patients) and are shown in blue (involve a between key transition). This is apparent from the high ratio of blue to red circles in the top left quadrant and indicates that the model takes a dip in inter-key intervals for between-hand-transition bigrams as a marker of PD.

A second property of key sequences that appears to be important is their transitional probability. There is good reason to think that PwPD will have difficulty deploying learned habits in typing. We know that the timing of keystrokes in typists is sensitive to the transitional probabilities between keys (Behmer and Crump, 2016), and we can take this as a marker of acquired habits. We would expect this relationship to be altered in PwPD. Mixed effects modelling with by-participant random intercepts and slopes confirms that this is the case in our data with an increase of a 26% of an IKI interquartile range for each unit of standard deviation in bigram surprisal (inverse log probability of each character given the previous character) for controls, and a significant 3% lower increase in PwPD ($p < 0.01$) across all participants. This indicates a reduced sensitivity to decreases in key transition probabilities in pwPD relative to controls. It is also the case that gradients are significantly higher for keys with high surprisal. Figure 8 displays the three-way relationship between gradient divergence, IKI divergence and surprisal and suggests that the model is picking up on the reduced effect of transitional probabilities on interkey intervals for PwPD relative to controls.

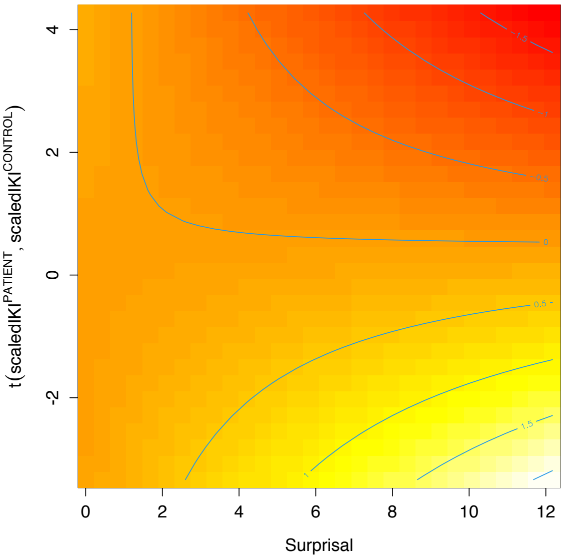


Figure 8: Contour – red (low) to yellow (high) – of gradient divergence (t-value for PwPD-Control comparison) for different values of scaled IKI divergence (t-value for PwPD-Control comparison again) and bigram surprisal. Gradient divergence is greatest for key transitions with high surprisal for which PwPD have low scaled IKI relative to controls. The model appears to pick up on the dampening of surprisal-related IKI spikes for PwPD relative to controls.

7 Conclusion

In this paper we have provided evidence that natural language processing techniques and in particular CNN-LSTM networks can identify markers of Parkinson's disease in logged typing behaviour. Critically there is good reason to think that the markers identified will have high specificity with regards to Parkinson's disease. While simple motor tests like the finger tapping test, and summary timing statistics from typing data, are widely used to distinguish PwPD from people without the disease, they rely on disease signs that PD has in common with other disorders - namely general slowing. In this work we first remove this disease sign from the data and then use a CNN-LSTM to pick up on more subtle changes in performance. We report very promising performance using this approach. We further report on an analysis of the gradients in our model which suggests that it is picking up on plausible effects of PD seen in the data.

Previous work has sought to distinguish PwPD from controls by observing how rapidly and consistently they press keys when typing. However, such work begins by discarding potentially valuable information - the identity of the keys pressed. We found that including key identify in our data/model provided a performance improvement relative to timing-only models. We found an improvement in performance (increased AUC) in identifying patients among participants tested in clinics in both English and Spanish. Furthermore we found a substantial leap in performance on the more difficult task of discriminating PD patients from controls in a new large dataset recruited and tested online.

These results suggest that NLP techniques allows us to identify theoretically-motivated markers of PD (Redgrave et al., 2010) in typing data. These incorporate both speed and character information, and so may be more robust than currently-used markers. Future work will of course require that we test this directly, by collecting typing data from people with other neurological disorders and using these markers for multi-class classification. This work is only the tip of the iceberg in terms of the contribution that NLP can make to the task of detecting signs of Parkinson's disease, and potentially other movement disorders, in typing data.

Acknowledgements

This work was supported by an MRC Confidence in Concept Award to the University of Liverpool.

References

Warwick R Adams. 2017. High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing. *PLOS ONE*, 12(11):e0188226.

Daniel Austin, Holly Jimison, Tamara Hayes, Nora Mattek, Jeffrey Kaye, and Misha Pavel. 2011. Measuring motor speed through typing: a surrogate for the finger tapping test. *Behavior Research Methods*, 43(4):903–909.

Colin Bannard, Mariana Leriche, Oliver Bandmann, Christopher H Brown, Elisa Ferracane, Alvaro Sánchez-Ferro, José Obeso, Peter Redgrave, and Tom Stafford. 2019. Reduced habit-driven errors in Parkinson's Disease. *Scientific Reports*, 9(1):3423.

Thomas R Barber, Johannes C Klein, Clare E Mackay, and Michele TM Hu. 2017. Neuroimaging in premotor Parkinson's disease. *NeuroImage: Clinical*, 15:215–227.

Lawrence P Behmer and Matthew JC Crump. 2016. Crunching big data with fingertips: How typists tune their performance toward the statistics of natural language. In *Big Data in Cognitive Science*, pages 329–345. Psychology Press.

Vanessa Buhrmester, David Münch, and Michael Arens. 2019. Analysis of explainers of black box deep neural networks for computer vision: A survey. *arXiv preprint arXiv:1911.12116*.

B Delisle Burns and J David DeJong. 1960. A preliminary report on the measurement of Parkinson's disease. *Neurology*, 10(12):1096–1096.

Francois Chollet. 2017. *Deep Learning with Python*, 1st edition. Manning Publications Co., USA.

Lonneke ML De Lau and Monique MB Breteler. 2006. Epidemiology of Parkinson's disease. *The Lancet Neurology*, 5(6):525–535.

Ray J Dolan and Peter Dayan. 2013. Goals and Habits in the Brain. *Neuron*, 80(2):312–325.

Julian M Fearnley and Andrew J Lees. 1991. Ageing and Parkinson's disease: substantia nigra regional selectivity. *Brain*, 114(5):2283–2301.

Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. 2016. How we type: Movement strategies and performance in everyday typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4262–4273.

Luca Giancardo, Alvaro Sanchez-Ferro, Teresa Arroyo-Gallego, Ian Butterworth, Carlos S Mendoza, Paloma Montero, Michele Matarazzo, José A Obeso, Martha L Gray, and R San José Estépar. 2016. Computer keyboard interaction as an indicator of early Parkinson's disease. *Scientific Reports*, 6:34468.

G Giovannoni, J Van Schalkwyk, VU Fritz, and AJ Lees. 1999. Bradykinesia akinesia incoordination test (BRAIN TEST): an objective computerised assessment of upper limb motor function. *Journal of Neurology, Neurosurgery & Psychiatry*, 67(5):624–629.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Fadila Hadj-Bouziane, Isabelle Benatru, Andrea Brovelli, Hélène Klinger, Stéphane Thobois, Emmanuel Broussolle, Driss Boussaoud, and Martine Meunier. 2013. Advanced Parkinson's disease effect on goal-directed and habitual processes involved in visuomotor associative learning. *Frontiers in Human Neuroscience*, 6:351.

Hasan Hasan, Maggie Burrows, Dilan S Athauda, Bruce Hellman, Ben James, Tom Warner, Thomas Foltynie, Gavin Giovannoni, Andrew J Lees, and Alastair J Noyce. 2019. The BRadykinesia Akinesia INcoordination (BRAIN) tap test: capturing the sequence effect. *Movement Disorders Clinical Practice*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Alastair J Noyce, Anna Nagy, Shami Acharya, Shahrzad Hadavi, Jonathan P Bestwick, Julian Fearnley, Andrew J Lees, and Gavin Giovannoni. 2014. Bradykinesia-Akinesia Incoordination Test: Validating an Online Keyboard Test of Upper Limb Function. *PLOS ONE*, 9(4):e96260.

Peter Redgrave, Manuel Rodriguez, Yoland Smith, Maria C Rodriguez-Oroz, Stephane Lehericy, Hagai Bergman, Yves Agid, Mahlon R DeLong, and Jose A Obeso. 2010. Goal-directed and habitual control in the basal ganglia: implications for Parkinson's disease. *Nature Reviews Neuroscience*, 11(11):760.

David R Roalf, Petra Rupert, Dawn Mechanic-Hamilton, Laura Brennan, John E Duda, Daniel Weintraub, John Q Trojanowski, David Wolk, and

Paul J Moberg. 2018. Quantitative assessment of finger tapping characteristics in mild cognitive impairment, Alzheimer's disease, and Parkinson's disease. *Journal of Neurology*, 265(6):1365–1375.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673 – 2681.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Michael Sharman, Romain Valabregue, Vincent Perlbarg, Linda Marrakchi-Kacem, Marie Vidailhet, Habib Benali, Alexis Brice, and Stephane Lehéricy. 2013. Parkinson's disease patients show reduced cortical-subcortical sensorimotor connectivity. *Movement Disorders*, 28(4):447–454.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.

A Preprocessing copy-typing data

In this section we outline the detailed steps that were taken in the preparation of the data for this study. As noted in the body, the datasets under consideration are those in table 1 which reproduce here for completeness.

Collection	Language	N_p^+/N_p^-	N_s^+/N_s^-
In-clinic	English	$16/25$	$426/739$
In-clinic	Spanish	$11/9$	$310/265$
Online	English	$99/130$	$1415/1862$

The following data-cleaning and data-wrangling steps were taken, to prepare the data for preprocessing:

1. Remove duplicate responses.

2. Calculate Levenshtein distance (edit distance) and remove sentences which have a measured value[3] above 75. This is done on the *typed* sentences (i.e. the ones seen by the user during the experiment, not the concatenated logged keys).

3. Remove all sentences where participants have employed ←, →, ↑ and ↓ keys. As the error-corrective behaviour becomes too-complex with their inclusion, they were removed to simplify the problem space.

4. Replace Space (spacebar) with a blank key to homogenise the dataset.

5. Make all sentences lower-case (to facilitate better inference in the modelling stage) – see (Zhang et al., 2015).

6. To create a homogeneous key corpus for all participants, the following keys (all were extracted from the dataset itself) were mapped to `<unk>`:

    - ContextMenu
    - Delete
    - End
    - Enter
    - F11
    - F16
    - \n
    - Home
    - Insert
    - MediaPreviousTrack
    - None
    - NumLock
    - PageDown
    - Process
    - Unidentified

    This is necessary because all participants took part in the data collection, used their own personal computer, and thus by extension their own keyboard. We use a US English keyboard which has a grid size of 5×14.

7. Remaining keys with a character length of more than one, are mapped to Greek letters so as to not corrupt the character encoding downstream in the NLM:

    - backspace → α
    - shift → β
    - control → γ
    - capslock → δ
    - meta → ϵ
    - tab → ζ
    - alt → η

8. We set an option which allows for the shift key to be completely dropped. We do this owing to its use for capitalising letters. As this is not of interest to us in this study we typically remove it.

9. Hold-time and inter-key interval outliers are removed and replaced with the first moment of a kernel density estimate of those timings, for all sentences typed by a given participant.

10. Backspace implementation is the next step, as described in §4.2.

[3] A cut-off value was selected by inspection, and it was found that any sentence which had a value below this was not informative enough to warrant inclusion.

B Model architecture

Table 4: Detailed description of NLM architecture

Layer	Details	
1D Convolution	#filters	16
	filter size	3
	stride	1
	L2 regularisation	1e-6
	Activation	ReLU
Dropout	probability	0.5
1D Convolution	#filters	8
	filter size	3
	stride	1
	L2 regularisation	1e-6
	Activation	ReLU
Dropout	probability	0.5
Bidirectional LSTM	#hidden units	64
	Activation	tanh
Fully connected	input size	64
	output size	2
	Activation	Softmax

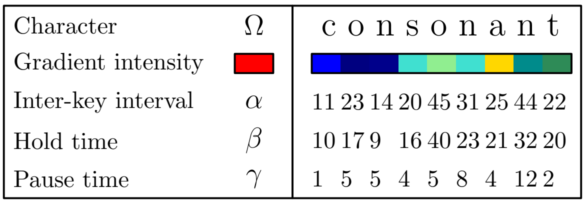
C Model training

To counter act over-fitting we use an array of standard techniques including dropout (Srivastava et al., 2014), weight regularisation and early stopping(Goodfellow et al., 2016, §7). Additionally we employ a task specific training schedule to aid feature learning in the convolutional layers. We first split our sentences into word pairs such that we effectively increase the number of samples, e.g. `Books include Penguin Island...`→`[Books include, Penguin Island,...]`. Given that the convolutional layers operate locally on the sentence we can then pre-train the filters on this augmented dataset in a more stochastic optimisation process. We then use the standard protocol (Chollet, 2017, §5.3) for transfer learning by freezing the convolutional filter weights before training ensues on the sentence datasets until performance on the validation set stops improving. Finally we unlock the convolutional filters and re-start training on the sentence dataset with a lower learning-rate and larger batch size.

The models are trained via Adam optimisation (Kingma and Ba, 2014) over shuffled mini-batches with early stopping terminating training if validation loss does not improve for 16 epochs. The initial learning rate is set to 0.001 and is decreased by a factor 0.5 if the validation loss does not improve for 10 epochs. We use a batch size of 16 for the word-pair convolutional filter pre-training and the first round of training on the sentence datasets, for second tuning round we increase the batch size to 32 and start with a learning rate of 10^{-4}. For regularisation we use dropout with probability 0.5 on and L2 regularisation with factor 10^{-6} on all convolutional layers.

D Additional Grad-CAM visualisations

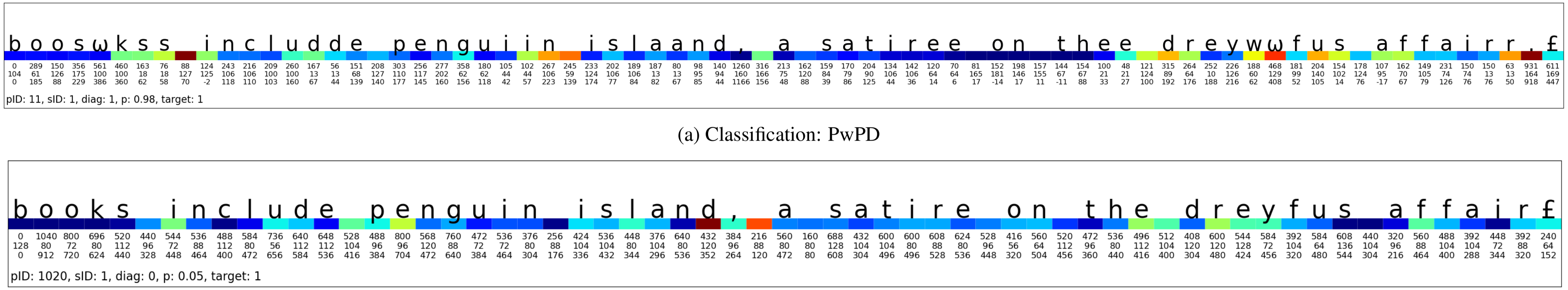
Shown in this section are additional examples of the 1D Grad-CAM, applied to all sentences, with positive and negative examples shown for each in the pages overleaf. The gradient and timing visualisations (the Grad-CAMs) can be understood by consulting the legend in fig. 9.



Character	Ω	c	o	n	s	o	n	a	n	t
Gradient intensity	🟥	🟦	🟦	🟦	🟩	🟩	🟩	🟨	🟩	🟩
Inter-key interval	α	11	23	14	20	45	31	25	44	22
Hold time	β	10	17	9	16	40	23	21	32	20
Pause time	γ	1	5	5	4	5	8	4	12	2

Figure 9: Grad-CAM legend. The *left panel* contains the keys and the respective magnitudes they measure. The *right panel* contains a small example of the 1D Grad-CAM applied to the word `consonant`.

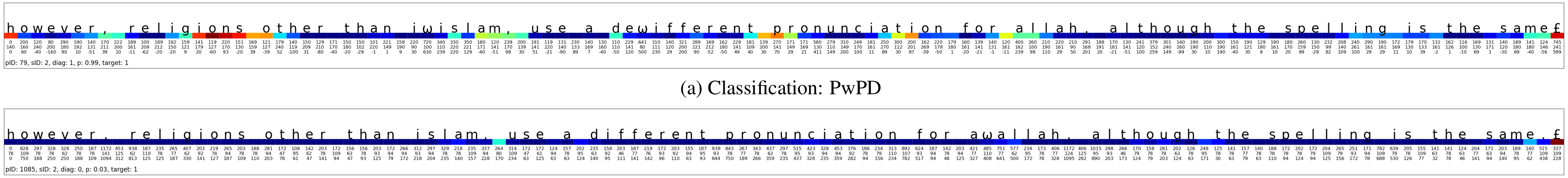
Additional Grad-CAM figures are shown overleaf.



(a) Classification: PwPD

(b) Classification: Control

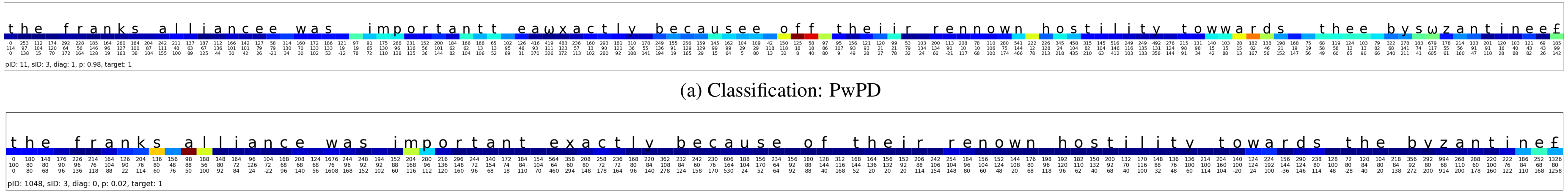
Figure 10: Sentence 1 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

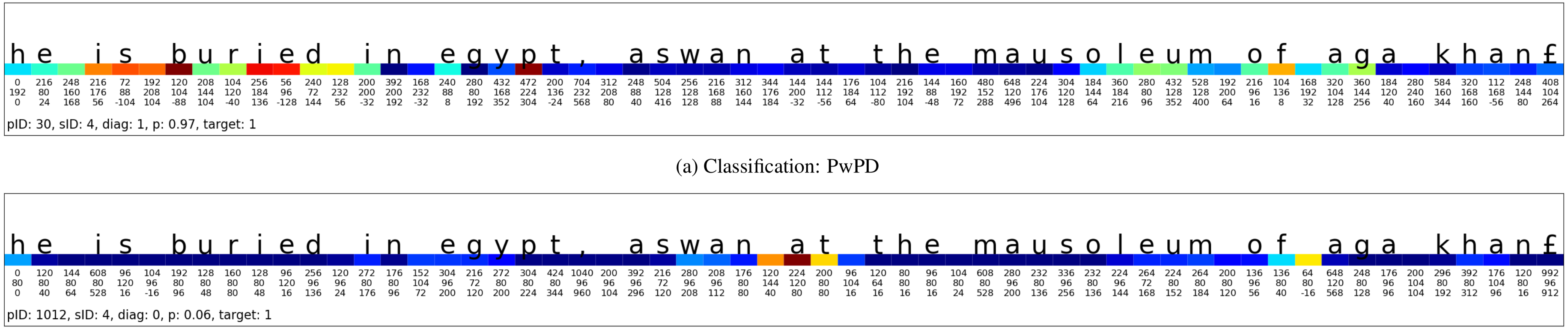
Figure 11: Sentence 2 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

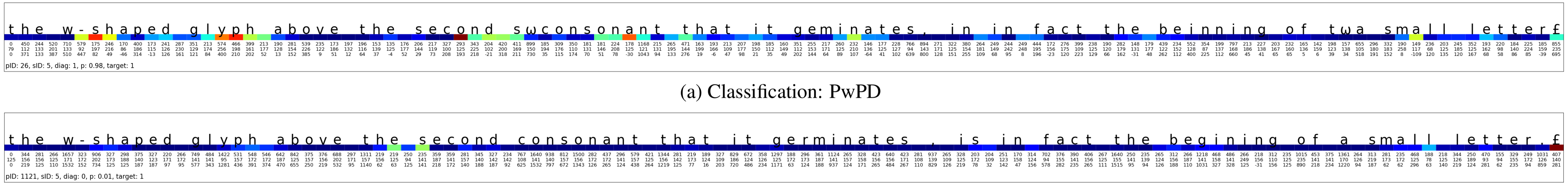
Figure 12: Sentence 3 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

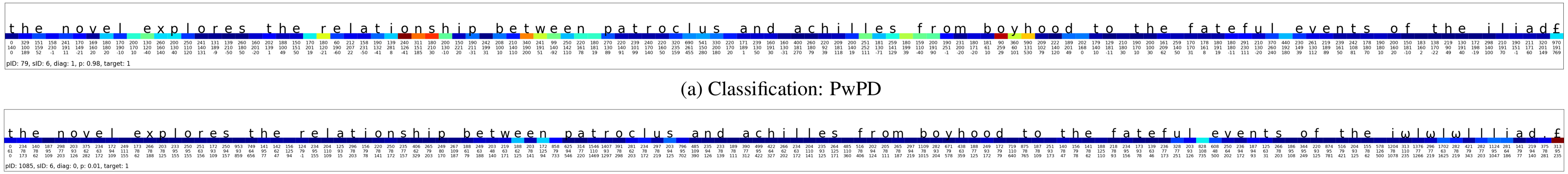
Figure 13: Sentence 4 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

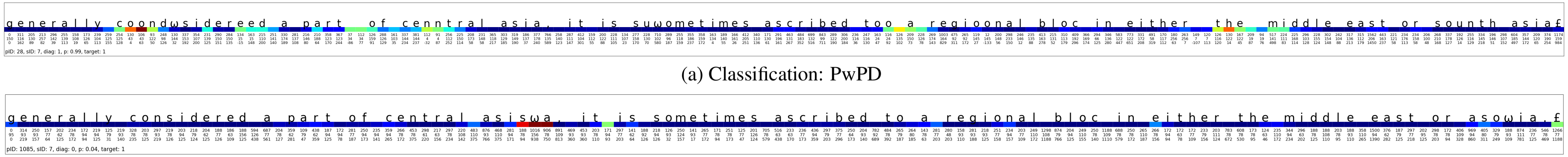
Figure 14: Sentence 5 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

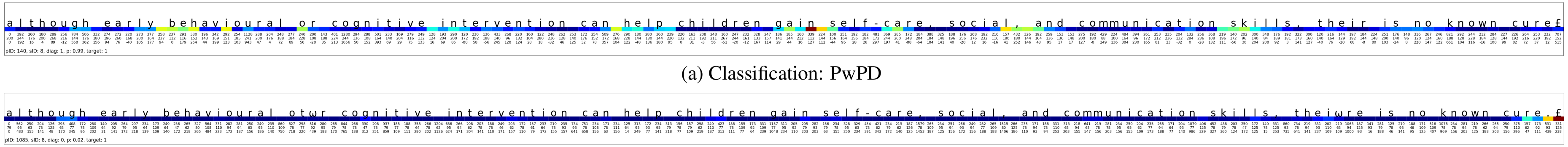
Figure 15: Sentence 6 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

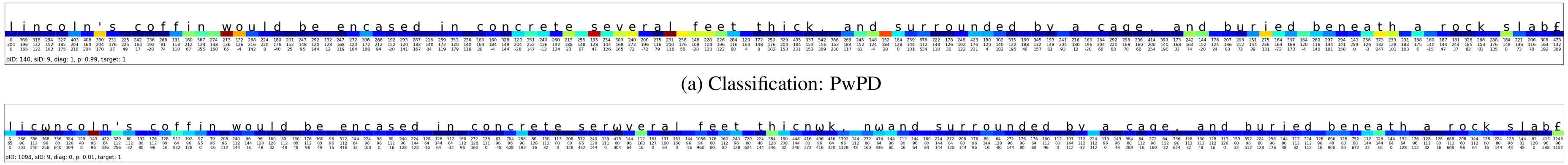
Figure 16: Sentence 7 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

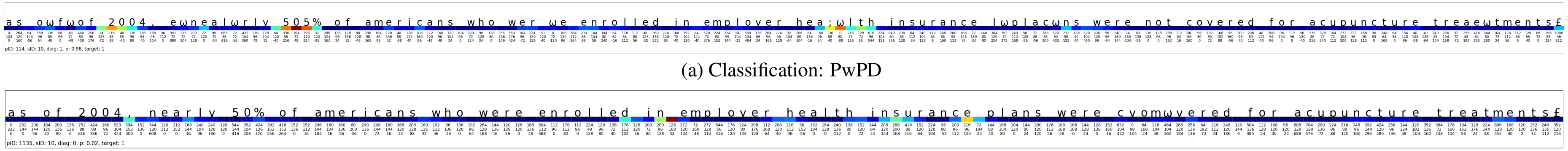
Figure 17: Sentence 8 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

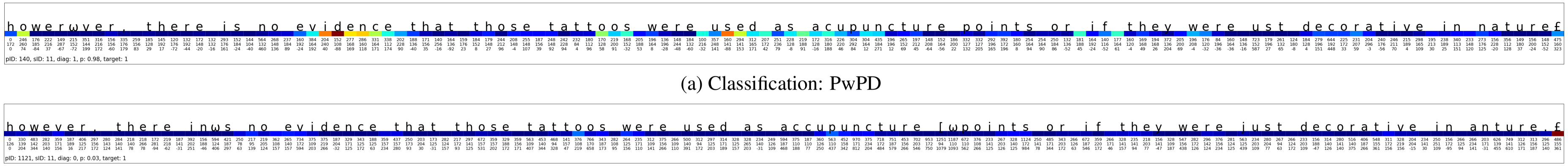
Figure 18: Sentence 9 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

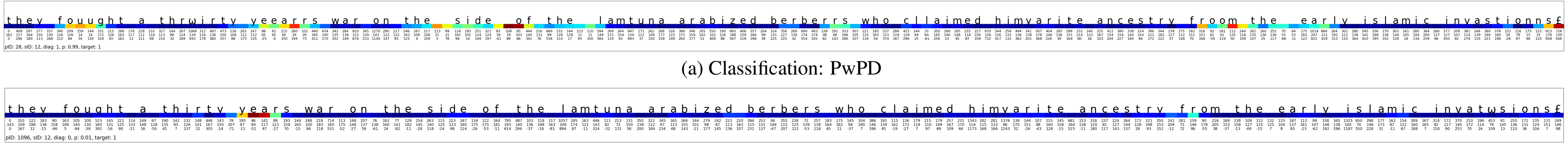
Figure 19: Sentence 10 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

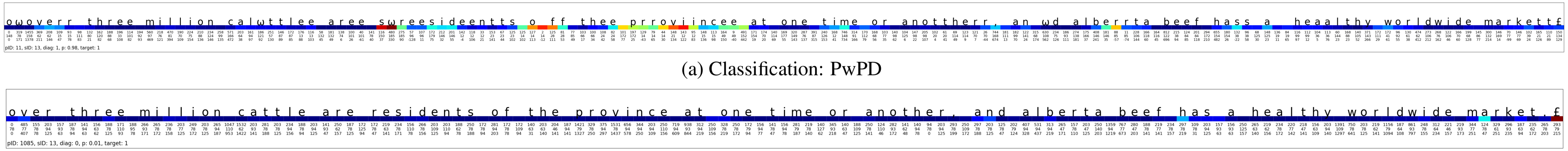
Figure 20: Sentence 11 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

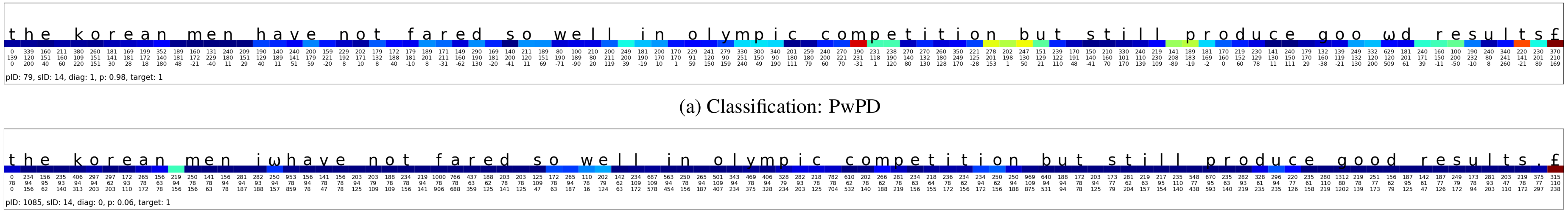
Figure 21: Sentence 12 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

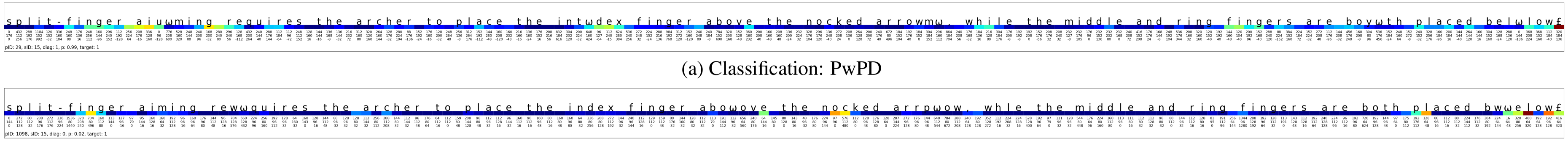
Figure 22: Sentence 13 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

Figure 23: Sentence 14 classification using the proposed approach. Positive and negative example shown.



(a) Classification: PwPD

(b) Classification: Control

Figure 24: Sentence 15 classification using the proposed approach. Positive and negative example shown.