From text saliency to linguistic objects: learning linguistic interpretable markers with a multi-channels convolutional architecture

Laurent Vanni, Marco Corneli, Damon Mayaffre, Frédéric Precioso

CORPUS

Corpus

24 | 2023
Les corpus numériques pour la didactique des langues : de la formation des enseignants à l'élaboration de dispositifs d'apprentissage

From text saliency to linguistic objects: learning linguistic interpretable markers with a multi-channels convolutional architecture

Des saillances du texte aux objets linguistiques : apprentissage des marqueurs linguistiques interprétables avec une architecture CNN multi-niveaux

Laurent Vanni, Marco Corneli, Damon Mayaffre and Frédéric Precioso

OpenEdition Journals

Electronic version
URL: https://journals.openedition.org/corpus/7667
DOI: 10.4000/corpus.7667
ISSN: 1765-3126

Publisher
Bases ; corpus et langage - UMR 6039

Electronic reference
Laurent Vanni, Marco Corneli, Damon Mayaffre and Frédéric Precioso, "From text saliency to linguistic objects: learning linguistic interpretable markers with a multi-channels convolutional architecture", *Corpus* [Online], 24 | 2023, Online since 15 January 2023, connection on 14 February 2023. URL: http://journals.openedition.org/corpus/7667 ; DOI: https://doi.org/10.4000/corpus.7667

From text saliency to linguistic objects: learning linguistic interpretable markers with a multi-channels convolutional architecture

Des saillances du texte aux objets linguistiques : apprentissage des marqueurs linguistiques interprétables avec une architecture CNN multi-niveaux

Laurent Vanni, Marco Corneli, Damon Mayaffre and Frédéric Precioso

1. Introduction

1 Each author has a discursive identity made up of identifiable lexical and grammatical choices. Therefore, one of the challenges of deep learning on text is to describe these identities.

2 Although it was shown in the literature that, in terms of accuracy, CNN based approaches outperform existing classifiers based on statistical key-indicators (e.g. the relative words frequency) or other machine learning techniques, it is still not clear if and how CNNs make use of standard features used in text mining (for instance word co-occurrences). We might also go further and assume that, for text classification, CNNs can rely on other complex linguistic structures that might be of interest for linguists. In the attempt to shed some light on this topic, our approach mainly relies on deconvolution process (i.e. transpose process), allowing us to interpret the CNN features in the input space.

3 This paper focuses on linguistic object analysis via a multichannel convolutional architecture. That is, a CNN is trained to associate several parts of transcribed political speeches to their speaker (e.g. E. Macron and D. Trump). Our main contribution is an improvement of an existing measure, the Text Deconvolution Saliency (TDS) (TDS, Vanni et al. 2018), called *weighted* Text Deconvolution Saliency (wTDS), allowing us to

visualize the linguistic markers used by the CNN to perform the classification of a text, but also to make them fully interpretable for the linguists. In order to have a relevant description of a dataset, the wTDS is included in a model that introduce two further contributions i) processing the CNN parameters in order to "rank" text segments assigned to an author from the more to the less representative of that author and ii) introducing a multi-channel CNN architecture in order to exploit additional linguistic information (e.g. lemma or *part-of-speech*) for each token.

4 The next section describes some of the most representative related works. Two of them are discussed in more details in order to motivate and better describe our own main contribution.

1.1 Related works

5 Since the seminal work of Collobert and Weston (2008), adopting CNNs for several NLP tasks (part-of-speech tagging, chunking, named entity recognition and semantic labeling), many researchers have widely used CNNs for similar and other purposes, such as text modeling (e.g. Kalchbrenner et al. 2014) or sentence classification (e.g. Kim 2014). While CNNs are not the only available deep architecture in Text Mining, it has been noticed that they have several advantages with respect to recurrent architectures (RNNs, in particular LSTM and GRU) when performing key-phrase recognition (Yin et al. 2017). This supervised classification task is the one we are interested in this work. In particular, we aim at uncovering linguistic patterns used to highlight *similarities* and *specificities* (Feldman & Sanger 2007, Lebart, Salem & Berry 1998) in a corpus. Standard text analysis techniques originally relied on statistical scores, for instance on the relative frequency of words (a.k.a. z-scores, see Lafon 1980). However, these techniques could not exploit more challenging linguistic features, such as syntactical motifs Mellet and Longrée (2009). In order to overcome these limitations and to account for long term dependencies in sentences, CNNs have been recently used. Indeed, being CNNs more robust than RNNs to the vanishing gradient problem, they might be able to detect links between different parts of a sentence (Dauphin et al. 2017, Wen et al. 2017, Adel & Schutze 2017). This property is crucial, since it was shown that long range dependencies emerge in real data (Li et al. 2015). Aiming at inspecting these dependencies as long as other complex linguistic patterns, some tools explaining how CNNs perform the classification task are required. In this regard, a recent crucial contribution is represented by the Local Interpretable Model-agnostic Explanations (LIME Ribeiro et al. 2016) framework. The basic idea of LIME is to approximate any complex classifier (e.g. a CNN) by a simpler one (e.g. sparse linear) in a neighborhood of a training point x_i. A simplified representation $\overline{x}_i$ of x_i is adopted, and N points in a neighborhood of $\overline{x}_i$ are sampled uniformly and used to minimize a distance between the original classifier and the simpler one. Once the simpler classifier is trained, it can be used to assess the (positive or negative) contribution of each feature to the classification task as easily as in linear models. This approach provides very interesting results and is generic, since it can provide explanations for any kind classifier. However, for every training point it involves sampling N neighbors and evaluating the classifier for each one of them. This might be computationally prohibitive, especially for high dimension data. In the context of key-phrase recognition, an alternative approach was proposed by Vanni et al. (2018). They considered as input data text segments of fixed size (M tokens). Each data point was represented as an $M \times D$ matrix,

where D is the word embedding size. After training a CNN for an author recognition task, they used a Deconvolution Network (Zeiler & Fergus 2014) to project the feature map back into the input data space. Thus, the "deconvolution" assigns to the m-th token in the i-th text segment (say d_{im}) a vector $x_{im} \in \mathbb{R}^D$. The sum of its entries defines the Text Deconvolution Saliency (TDS) of d_{im}. Intuitively, the higher (respectively lower) the TDS of d_{im}, the more (less) d_{im} contributed to assign the text segment to its class (i.e. its author). Although this approach returns meaningful results it may suffer from some inconsistencies in the explanation, as it will be shown in Section 2. In order to preserve the computational efficiency of TDS (once the CNN is trained it can be computed at a cost of one model evaluation per data point) we propose an improved version of the TDS (Section 2.2) overcoming the explanation drawbacks.

6 This paper is organized as follows: Section 2 describes our CNN architecture as well as our contributions. Section 3 illustrates the framework described in Section 2 on two datasets: a English corpus and a French corpus. Section 4 concludes the paper and outlines some perspectives for future research.

2. Model and contributions

7 The first part of this section details our model, a convolutional neural network, trained for author classification tasks. In this work, this task corresponds to an intermediate step but does *not* represent our final goal. Indeed, the scope is to learn how to exploit a trained CNN to recover linguistic markers, specific to the different authors. Thus, after detailing the architecture, we focus on some original contributions to the linguistic features extraction. Our main contribution, the *weighted Text Deconvolution Saliency* (wTDS) is described in Section 2.2. Two other contributions, the *softmax breakdown ranking* and the *multi-channel convolutional lemmatization* are discussed in Section 2.3.

8 **Notation.** In the following, $v \in \mathbb{R}^N$ will denote a real vector v with N entries. If not differently stated, it is intended to be a column vector. The notation $A \in \mathbb{R}^{M \times N}$ will be used to define a real matrix with M rows and N columns and the function $relu(\cdot)$ is defined as

$$relu(x) = max\{0, x\}.$$

2.1 CNN baseline

9 The CNN considered takes as input $d_1,...,d_N$ text segments, each containing a fixed number of tokens M. In the examples that we consider in Section 3 each segment is part of a presidential speech, so that the number of classes K is the number of considered presidents. An embedding layer is used for word representation. Although this layer might rely on different well known models such as fastText (Bojanowski et al. 2017, Joulin et al. 2017), Word2Vec (Mikolov et al. 2013) or Glove (Pennington et al. 2014) as long as a fine tuning of the embedding vectors is allowed during optimization, the choice of the embedding model is not crucial. Once the word feature vectors are obtained, they are concatenated (by row) in such a way to form a matrix with M rows. This resulting matrix can then be input into a convolutional layer applying several filters all having the same width as the dimension of the embedding matrix. One max-pooling layer follows, equipped with a nonlinear activation function. A deconvolutional layer (up-sampling plus convolution with transpose filters) is then introduced to bring

the convolutional features back into the word embedding space. Finally, two fully-connected layers and a *softmax* function output for each segment d_i a vector $\hat{z} \in \{0, 1\}^K$, where K is the number of classes/authors. The following multinomial cross-entropy loss function is considered:

$$\mathcal{L}(\theta) := -\sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \log\left(\hat{z}_{ik}(\theta)\right) \tag{1}$$

where θ denotes the set of all the network trainable parameters and $z \in \mathbb{R}^{N\times K}$ is an observed binary matrix, whose k-th row encodes the class/author of the i-th text segment (thus $z_{ik} = 1$ iff d_i is affected to the k-th class/author). The above loss function is minimized with respect to θ via an Adam optimizer. In order to avoid overfitting the whole dataset is split into train (80%) and validation (20%) sets and the loss function in Eq. (1) is monitored on the validation set during optimization, allowing us to apply early stopping (Prechelt 1998) (Figure 1). A graphical representation of the model described so far can be seen in Figure 2.

Figure 1. Model loss and accuracy

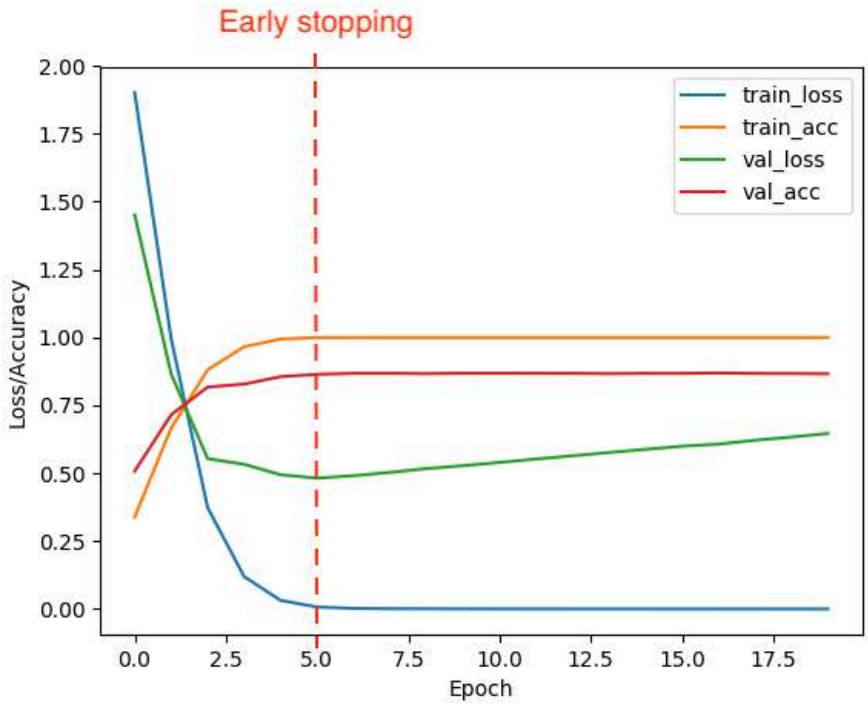
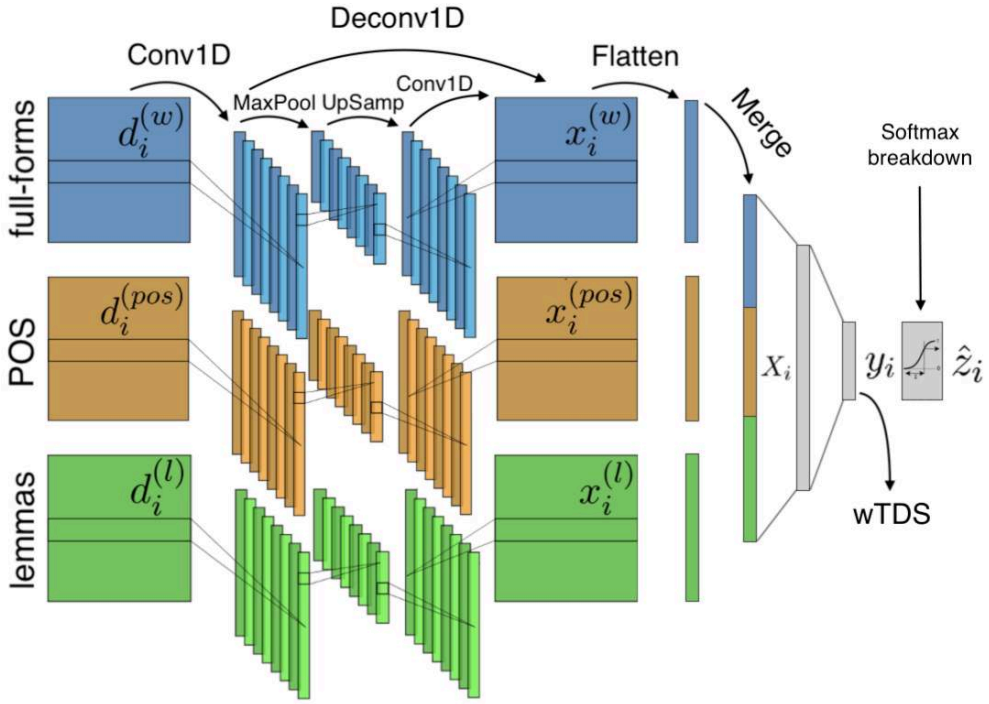


Figure 2. Three channels convolution/deconvolution for three representation of the input 1) *full-forms* (words), 2) *part-of-speech* (POS), 3) *lemma*



2.2 A new enriched TDS

11 After the CNN has been trained on the train dataset, it can assign a text segment d_i (either in the train or in the validation set) to its class/author. We recall that d_i can be viewed as a real matrix with M rows, where M is the number of tokens of di and D columns, where D is the embedding size. The m-th token of d_i, corresponding to the m-th row of the matrix, is denoted by d_{im} and it is a vector in $\mathbb{R}^D$. The deconvolutional layer (see Figure 2) assigns to every d_{im} another vector of the same size denoted by $x_{im} \in \mathbb{R}^D$. Note that, since this representation is the output of two convolutional layers, it is sensitive to the context of d_{im} (neighbor tokens). The Text Deconvolution Saliency (TDS, Vanni et al. 2018) of the token d_{im} is defined as

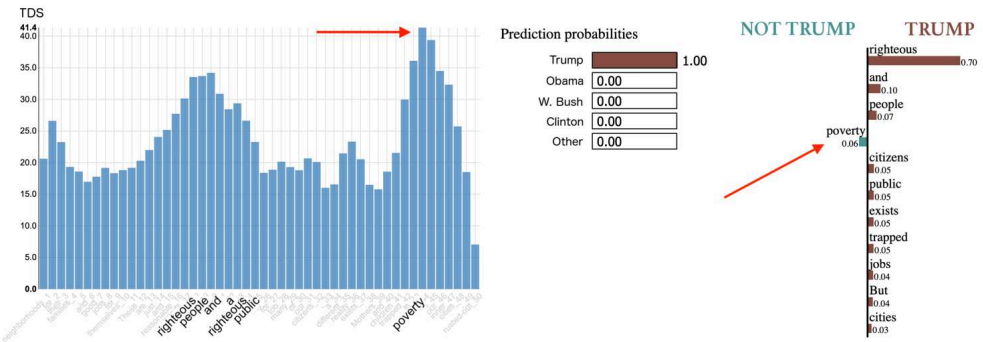
$$TDS(d_{im}) = \sum_{d=1}^{D} x_{imd} \tag{2}$$

12 where the real number x_{imd} the d-th entry of x_{im} We stress that, although this measure is defined for each token of d_i it also accounts for the context of d_i (see also the experiments in Section 3). The authors in Vanni et al. (2018) argue that, the higher the TDS of a token, the more the token (conditionally to its context) plays a crucial role in the classification task, according to the CNN. As a matter of fact, even though TDS can correctly highlight the relevant words/contexts in d_i being used by the CNN to classify d_i, it cannot tell us *how* the network uses them. To illustrate this point in more detail, consider the following extract from a speech by Donald Trump:

> […] neighborhoods for their families, and good jobs for themselves. These are just and reasonable demands of righteous people and a righteous public. But for too many of our citizens, a different reality exists: Mothers and children trapped in poverty in our inner cities; rusted-out […]
> (D. Trump, the 20th of January 2017, Inaugural Address, United States Capitol Building in Washington, DC).

13 This text is part of a corpus described in Section 3 and collects several part of speeches from the US presidents. Once properly trained for an author recognition task, the CNN detailed in the previous section can correctly recognize this speech as being pronounced by the president Trump.

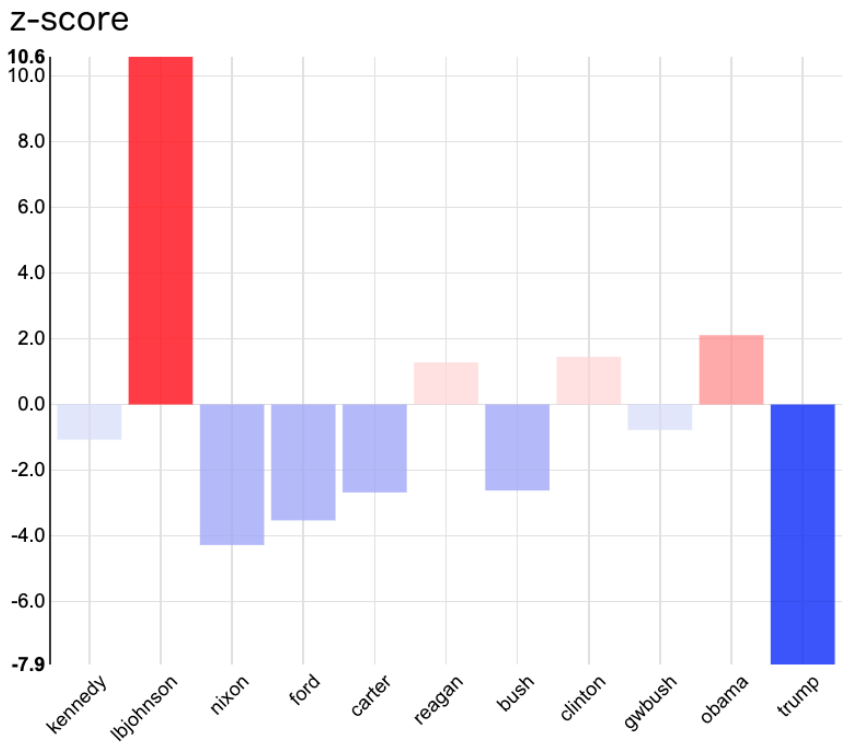
Figure 3. Comparing the activation boost of the tokens toward the class "Trump" according to TDS and LIME



(a) TDS (b) LIME

14 In Figure 3a a histogram reports the TDS scores for the tokens of the extract. The higher the bars, the more the corresponding tokens had a key role in the classification task. Now, when comparing these TDSs with the word contributions detected by LIME (Figure 3b) we see that most of the tokens having a high TDS correspond to brown right bars having a *positive* impact in classifying the speech as "Trump" (e.g. righteous, people). Conversely, according to LIME, the noun "poverty" seems to have a negative boost when performing a binary classification "Trump" or "No Trump". Indeed, if we additionally compute the z-scores of the tokens of d_i (Figure 4), with respect to the whole corpus, we see that the noun "poverty" is underused by D. Trump and this is in line with the explanation provided by LIME. However, this noun is very specific to another president in the corpus: L.B. Johnson. Thus, the importance of the word "poverty" was correctly captured by TDS, but we cannot say if that word contributed *for* "Trump" or *against* "Trump".

Figure 4. Z-scores for the noun “poverty” for the US presidents in the analyzed corpus



15 This motivated us to improve the TDS score initially proposed by Vanni et al. (2018), with two additional features: i) it should be able to go negative to indicate negative contributions of words to some classes and ii) in case of multi-class classification, for a word d_{im} it should be able to quantify its contribution to each class. In order to build such a measure, note that the last two fully connected layers of the CNN basically map the de-convolved features $x_{i1},\ldots,x_{iM}$ into a single vector in $\mathbb{R}^K$, denoted y_i (see Figure 2), where K is the number of classes. If we concatenate $x_{i1},\ldots,x_{iM}$ into a column vector X_i, of size $D{\times}M$, the map can be specified as

$$y_i = d + C\left(\text{relu}\left(b + AX_i\right)\right) \tag{3}$$

16 where $A \in \mathbb{R}^{E\times DM}$, $b \in \mathbb{R}^E$, $C \in \mathbb{R}^{E\times K}$ and $d \in \mathbb{R}^K$ and E is the size of the penultimate layer. In order to obtain a score that is specific to the token d_{im} we observe that

$$AX_i = \sum_{m=1}^{M} A_m x_{im}^T \tag{4}$$

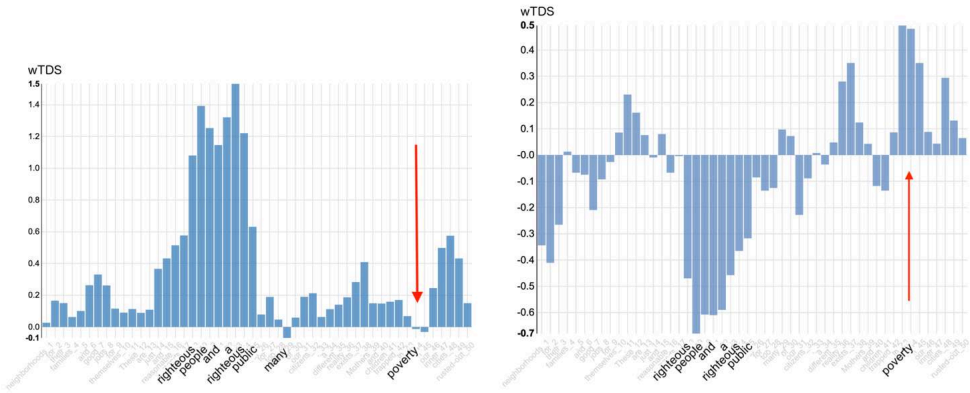
17 where $A_m \in \mathbb{R}^{E\times D}$ is the sub-matrix of A obtained by selecting all the rows and the D columns form the $(D(m-1)+1)$-th to the $(D(m-1)+D)$-th. Thus we define

$$wTDS(d_{im}) := d + C\left(\text{relu}\left(b + A_m x_{im}^T\right)\right) \tag{5}$$

18 Note that, instead of $TDS(d_{im})$, $wTDS(d_{im})$ is a vector with K entries. Each entry quantifies the activation boost of word d_{im} (conditionally to its context) for the class K. Moreover, the matrix multiplication $A_m x_{im}$ induced K *weighted* sums of the entries of x_{im}, in contrast

with the simple sum defined in Eq. (2). For this reason we call the measure in Eq. (5) **weighted Text Deconvolution Saliency (wTDS)**. Figure 5 shows the wTDSs for the class "Trump" of the tokens in the Trump's speech reported above. As it can be seen, the word "poverty" now has a small negative contribution when classifying the speech as "Trump". We notice that, once the CNN is trained, the computation of the wTDS for one token (for all the classes) has the cost of the matrix multiplications in Eq. (5). This is a huge advantage compared to LIME for two reasons: first, no sampling is required.

Figure 5. wTDS for classes "Trump" and "Johnson" for the tokens in the sample speech of D. Trump



(a) Trump (b) Johnson

19 Second, whereas LIME can only provide us with the tokens contribution in the binarized problem (e.g. "Trump" vs. "No Trump"), wTDS computes the tokens contribution to each class in one shot.

2.3 Softmax breakdown ranking

20 In the previous section, we described how, given an input text segment d_i, wTDS can be used to assess the contribution of each token in *di* for the class assignment. Now, we zoom one step out and try to *detect* the *key-segments* in the data set, i.e. the segments being the more representative of each author according to the CNN. In particular, it might be of interest to be able to rank $d_1,\ldots,d_N$ from the most to the least representative for each author.

21 A possible way to do that is described in the following. The number of neurons in the last layer of the deep CNN coincides with the number of classes, previously denoted by K. In the previous section $y_i \in \mathbb{R}^K$ denoted the value of that layer for the text segment d_i. Thus, y_{ik} is the value of the k-th neuron and it is a real number. As usually, a *softmax* activation function is applied to y_i in such a way to obtain K probabilities $\hat{z}_{ik}$ (see Figure 2) lying in the $K-1$ simplex

$$\hat{z}_{ik} = \frac{\exp(y_{ik})}{\sum_{j=1}^{K} \exp(y_{ij})} \tag{6}$$

22 Note that the above $\hat{z}_{ik}$ is the very same as in Eq. (1). The highest probability $\hat{z}_{ik}$ corresponds to the class assigned by the network to the observation d_i. However, if one entry of y_i is significantly higher than the others, it is mapped to 1 by the *softmax* transformation and all the other entries are mapped to zero. For instance, consider two

de-convolved features y_i and y_j corresponding to two different documents both assigned to class k. Assume also that $y_{ik} > y_{jk}$, so that the document d_i is more representative of the class than d_j. If y_{ik} and y_{jk} are large enough, after applying the *softmax* function they both will be mapped to one and it will no longer be possible to assess whether d_i or d_j is more representative of class k. Thus, we make unconventional use of the trained deep neural network and observe the activation rate of neurons *before* applying the *softmax* transformation. Doing that, allows us to sort the learning data (text segments) based on their activation strengths. This simple but efficient method provides us with the most relevant key-segment in the corpus for each class.

2.4 Multichannel convolutional lemmatization

23 Often, CNN for images have multiple channels. Indeed, the RGB colors encoding could be considered as three different representations of the input. Each representation corresponds to a data matrix and the convolutional layers apply different filters to each matrix and then later merge the results. Also with texts, it is possible to encode the data in multiple channels that might be used, for instance, to combine different word embedding solutions (skip-gram, cBow or Glove). Apart from word embedding, a pre-tagging process (Collobert & Weston 2008) allows data scientists and linguists to get supplementary material on each word, such as the *part-of-speech* (POS) and the *lemma*. Both of them are essential for a linguistic interpretation of the key-segments and to observe complex linguistic patterns (a.k.a syntactical motifs Mellet & Longrée 2009). It is those reasons motivated us to implement a multi-channel CNN to account for the POS and the lemma. However, using a single multi-channel convolutional layer to learn those patterns from each representation is not convenient for our purposes. Indeed, the max pooling operations merge all the information into one channel, thus making it impossible to retrieve which representation (word, POS or lemma) contributed to the classification. Since the aim of our contribution is to interpret the classifier, we split the convolution (and the max pooling) in three parts, one for each channel (see Figure 2). By doing that, the deconvolution mechanism can be applied to the three channels separately and all the linguistic features can be observed right after the deconvolutional layers. Finally, to combine this information, the features are merged into a global vector and the final dense layers use them to perform the class assignment. In more details, the m-th token of the segment d_i is now represented by three embedding vectors, say $d_{im(w)}$ for the full form, $d_{im(pos)}$ for the POS and $d_{im(l)}$ for the lemma (see Figure 2). After deconvolution, these embedding vectors are mapped to $x_{im(w)}$, $x_{im(pos)}$ and $x_{im(l)}$, respectively. Thus, whereas with a single channel, $wTDS(d_{im})$ was a vector in $\mathbb{R}^K$, in a multichannel environment, we can define **three** wTDS vectors in $\mathbb{R}^K$ for each token. For instance, $wTDS(d_{im(l)})$ refers to the lemma component of the m-th token and it can be computed as

$$wTDS(d_{im}^{(l)}) := d + C\left(\mathbf{relu}\left(b + A_m^{(l)}(x_{im}^{(l)})^T\right)\right)$$

24 where $A_{im(l)}$ denotes a sub-matrix accounting for the lemma channel (the green one in Figure 2) *and* the m-th token $x_{im(l)}$.

3. Experiments

25 Political discourse analysis is one of the major challenges for linguistics in textual data analysis. For many years, statistics have provided tools and results that help linguists to interpret political speeches. We will now see how our deep architecture allows us to describe international political discourses. We propose to test our model by analysing two political discourse corpora in two different languages, English and French. For comparison reasons, these two corpora are made from presidential speeches and respect the same chronological span, from the 1960s to today.

26 The first dataset targets American political discourse. It is a corpus of 1.8 millions of words of American presidents from J.F. Kennedy in 1961 to D. Trump in 2019. With 11 presidents, we focus on D. Trump to make a short but profound linguistic analysis of the discourse of the current US president. The second is symmetrical with the speeches of the French presidents under the 5th republic from 1958 to today. It is 8 French presidents from C. De Gaulle to E. Macron with 2.7 millions of words we focus also on current president, E. Macron.

27 By default, the accuracy of each model (English and French) exceeds 90%, but the markers displayed by the wTDS seem to be too sensitive to low frequencies (very rare linguistic markers) or on the contrary very frequent but unique to a president (high z-score). The purpose of our architecture being to observe new linguistic markers different from those known by statistics, each corpus has been filtered with precise rules to reduce the weight of these markers. Some words have been replaced: i) proper names ii) dates iii) words only present in a president. These rules reduce model accuracy by about 10% but help to reduce overfitting and extract relevant key segments. The table 1 compare those models, unfiltered (English, French) and filtered (English*, French*).

Table 1. English and French datasets

dataset	authors	vocab	words	acc
English	11	33279	1 815 839	90%
English*	11	14758	1 815 839	81%
French	8	46978	2 738 652	91%
French*	8	20211	2 738 652	84%

3.1 English data set

28 Section 2.2 introduce a key-segment of D. Trump detected with the *softmax break-down ranking* method with a simple model using only one channel for the full-form of words. With the *multi-channel convolutional lemmatization* (Section 2.3), we have now a wTDS score on each token for each channel and this selected segment become fully interpretable for the linguists due to exploitable features on full-form (blue words), *part-of-speech* (orange words) and lemma (green words):

> […] neighborhoods for their families, and good jobs for themselves. These are just and reasonable demands of righteous people and a righteous public SENT But for too many of our citizens, a different reality exists: Mothers and children trapped in poverty in our inner cities; rusted-out […]

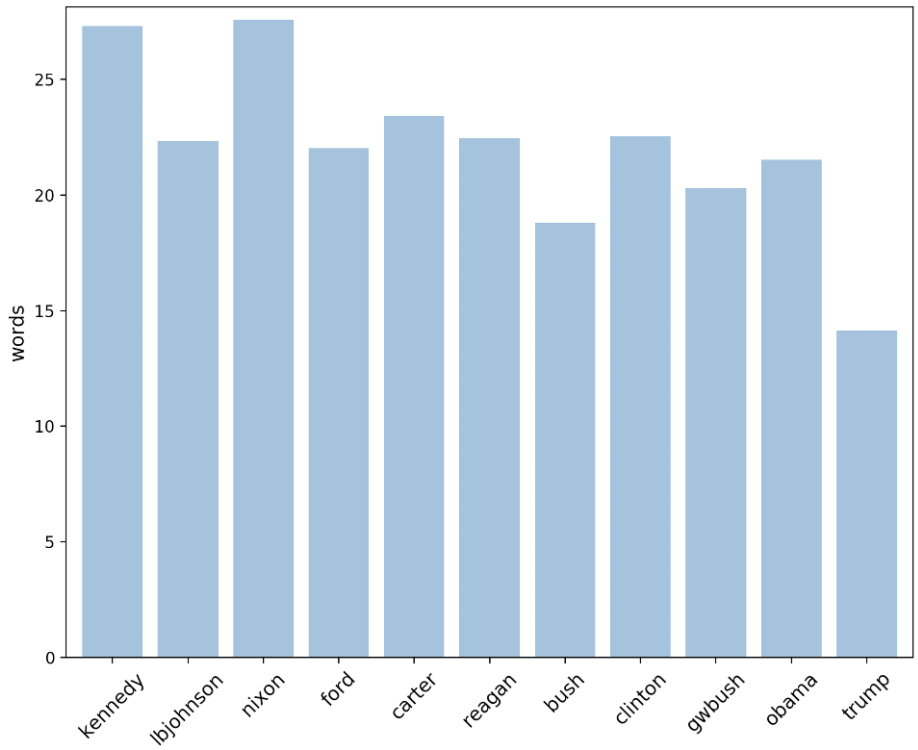
29 We highlight here the main activation zones having a wTDS higher than a fixed threshold. As it can be seen, there is a redundancy of "righteous people" and "righteous public", being part of a simple and compassionate vocabulary (e.g. "families", "mothers", "children" or simply "good jobs"), which is typical of populist speeches.

30 "But" appears as a characteristic of a polemical discourse that defines Trump's rhetoric. The president rarely makes a consensual speech. Opposition marks, as "But", allow him to build a speech setting him apart from the mainstream. Being "But" placed at the beginning of the sentence, its full-form wTDS highlights the role of conjunction of opposition rather than of conjunction of coordination.

31 We also report that the full-form wTDS for the word "many" is negative (Figure 3a). Since "many" is one of the words more often employed by president Trump (high z-score), a negative wTDS might appear surprising. However in this context, "many" is preceded by "too" which is taken into account by the convolution layer. Thus we checked the z-score of the linguistic pattern "too many", and we found out that it is higher for B. Obama than D. Trump. This is a very good example of the wTDS capability to capture the linguistic context.

32 Finally, the wTDSs of *part-of-speech* focuses on a simple but essential marker, the dot (encoded as "SENT"). The over use of this marker refers to a fundamental rhetorical choice of D. Trump: short sentences. The reduction of the sentence length is a trend that can be observed in most democracies in Europe or in USA. In the attempt to be accessible to as many people as possible, D. Trump's speech thus plays on syntactic simplification (Norris & Inglehart 2019). For a long time, political discourse has imitated literature with long sentences and relative or subordinate proposals, but nowadays, political discourse imitates popular language with short sentences that include only one subject, one verb and one complement. On average, in the corpus, Trump's sentence counts 14.15 words when Obama's sentence counts 21.51 words (Figure 6). In fact, the end of sentence markers characterize the current president. In 50 words here, Trump seems to take up the linguistic characteristics of populist discourse (Oliver & Rahn 2016) as it is expressed in the United States and Europe at the beginning of the 21st century.

Figure 6. Average sentence size



3.2 French data set

33 This section aims at demonstrating that Deep learning can easily adapt to the subtleties of each language. A French presidential corpus is considered. In this dataset, the segment that the model identifies as being the most characteristic of E. Macron's speech gathers remarkable features of the current French president language. The wTDSs highlight linguistic markers with multiple interpretations:

> […] intérêts industriels et qui construire le opacité PRP PRP:det décisions collectives qu'attendent nos concitoyens. La cinquième clé de notre souveraineté passe par le numérique. ce défi est aussi celui d' une transformation profonde de nos économies, de nos sociétés, de notre imaginaire même. La […]
> (Macron, the 26th of September 2017, speech about Europe at the Sorbonne)

34 Some main features of the E. Macron's speech emerge. First, the French president tries to give a non-ideological and pragmatic talk oriented towards action, movement and efficiency (Colen 2019). Thus, the lemmas "construire" (to build) and "transformation" are very meaningful of such a discourse whose main scope is to be dynamic. The word "numérique" (digital) is often at the heart of the speech of a president who talks about changes and who wants to show his technical modernity. Then, from a grammatical and syntactic point of view, most of the time, the "PRP PRP:det" sequence (meaning preposition + contracted article, in French) introduces adverbial phrases. Thus, E. Macron avoids the main topics but he is precise with the modalities of the action. In E. Macron's speech, both the subject and the object are less important than the way of the proposed reforms. Finally, from a lexical point of view, the CNN seems to focus on "concitoyens" (fellow citizens) which allows E. Macron to avoid the term "compatriots", considered too nationalist in the 21st century, in the context of the

European integration. A high wTDS also corresponds to the “nos” and “notre” (“our” and “ours”) forms as well to the lemma “notre”. Indeed, the construction of a political “we” appears as the main rhetorical objective of a discourse that aims at gathering the people behind its leader.

4. Conclusion and perspectives

35 We have introduced and tested a new method to extract relevant linguistic objects characterizing the different classes/authors in a multi-class classification context. The main focus of the present work are the hidden layers of a trained CNN. In particular we introduced a measure (wTDS) which, entirely relying on the learned parameters, allowed us to detect the key words that, conditionally to their context, were used by the CNN to assign a text segment to its author. We have proposed a routine to rank the text segments from the most to the least representative for each author providing a new and different view in the author discourse analysis. The way we propose to compute all these features internally to the network leads to a highly reduced computation cost (compared to LIME for instance) and thus allows us to design a multi-channel architecture accounting for *part-of-speech* and the lemma leading to extract enriched linguistic objects at almost no cost. The linguistic objects that we learn in this multi-class classification framework are those better discriminating one author *with respect to* the others. In order to extract not only discriminative spatial linguistic objects (using CNNs) but to take into account the sequential generation of the discourse based on these linguistic objects, recurrent networks have to be considered. Some tools already explore the hidden layers of such architectures (e.g. LSTMVis[1]) and future works might focus on the combination of both approaches, for instance, first extracting spatial patterns then analyzing their sequential organization for an even more in depth discourse analysis.

BIBLIOGRAPHY

Adel H. & Schutze H. (2017). “Global normalization of convolutional neural networks for joint entity and relation classification”, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1723-1729.

Bojanowski P., Grave E., Joulin A. & Mikolov T. (2017). “Enriching word vectors with subword information”, *Transactions of the Association for Computational Linguistics* 5: 135-146.

Cole A. (2019). *Emmanuel macron and the two years that changed France*. Manchester: Manchester University Press.

Collobert R. & Weston J. (2008). “A unified architecture for natural language processing: Deep neural networks with multitask learning”, in *Proceedings of the 25th International Conference on Machine Learning, ICML ’08.* New York: ACM, 160-167.

Dauphin Y. N., Fan A., Auli M. & Grangier D. (2017). “Language modeling with gated convolutional networks”, in *International Conference on Machine Learning*, 933-941.

Feldman R. & Sanger J. (2007). *The Text Mining Handbook. Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.

Joulin A., Grave E. & Mikolov P. B. T. (2017). “Bag of tricks for efficient text classification”, *EACL 2017*, 427.

Kalchbrenner N., Grefenstette E. & Blunsom P. (2014). “A convolutional neural network for modelling sentences”, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 655-665.

Kim Y. (2014). “Convolutional neural networks for sentence classification”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746-1751.

Lafon P. (1980). “Sur la variabilité de la fréquence des formes dans un corpus”, *Mots* 1(1): 127-165.

Li J., Chen X., Hovy E. & Jurafsky D. (2015). “Visualizing and understanding neural models in nlp”, *arXiv preprint arXiv:1506.01066*.

Lebart L., Salem A. & Berry L. (1998). *Exploring Textual Data*. Dordrecht: Springer.

Mayaffre D. & Vanni L. (eds.) (2021). *L’intelligence artificielle des textes. Des algorithmes à l’interprétation*. Paris: Honoré Champion.

Mellet S. & Longrée D. (2009). “Syntactical motifs and textual structures”, *Belgian Journal of Linguistics* 23: 161-173.

Mikolov T., Sutskever I., Chen K., Corrado G. S. & Dean J. (2013). “Distributed representations of words and phrases and their compositionality”, in *Advances in neural information processing systems*, 3111-3119.

Norris P. & Inglehart R. (2019). *Cultural backlash: Trump, brexit, and authoritarian populism*. Cambridge: Cambridge University Press.

Oliver J. E. & Rahn W. M. (2016). “Rise of the trumpenvolk: Populism in the 2016 election”, *The ANNALS of the American Academy of Political and Social Science* 667(1): 189-206.

Pennington J., Socher R. & Manning C. (2014). “Glove: Global vectors for word representation”, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.

Prechelt L. (1998). “Early stopping-but when?”, in *Neural Networks: Tricks of the trade*. Dordrecht: Springer, 55-69.

Ribeiro M. T., Singh S. & Guestrin C. (2016). “Why should i trust you?: Explaining the predictions of any classifier”, in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135-1144.

Vanni L., Ducoffe M., Aguilar C., Precioso F. & Mayaffre D. (2018). “Textual deconvolution saliency (tds): a deep tool box for linguistic analysis”, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 548-557.

Wen T.-H., Vandyke D., Mrksic N., Gasic M., Barahona L. M. R., Su P.-H., Ultes S. & Young S. (2017). “A network-based end-to-end trainable task-oriented dialogue system”, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1: 438-449.

Yin W., Kann K., Yu M. & Schutze H. (2017). *Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923.*

Zeiler M. D. & Fergus R. (2014). “Visualizing and understanding convolutional networks”, in *European conference on computer vision*. Dordrecht: Springer, 818-833.

NOTES

1. http://lstm.seas.harvard.edu/

ABSTRACTS

A lot of effort is currently made to provide methods to analyze and understand deep neural network impressive performances for tasks such as image or text classification. These methods are mainly based on visualizing the important input features taken into account by the network to build a decision. However these techniques, let us cite LIME, SHAP, Grad-CAM, or TDS, require extra effort to interpret the visualization with respect to expert knowledge. In this paper, we propose a novel approach to inspect the hidden layers of a fitted CNN in order to extract interpretable linguistic objects from texts exploiting classification process. In particular, we detail a *weighted* extension of the Text Deconvolution Saliency (wTDS) measure which can be used to highlight the relevant features used by the CNN to perform the classification task. We empirically demonstrate the efficiency of our approach on corpora from two different languages: English and French. On all datasets, wTDS automatically encodes complex linguistic objects based on co-occurrences and possibly on grammatical and syntax analysis.

De nombreux efforts sont actuellement déployés pour fournir des méthodes d'analyse et de compréhension des performances remarquables des réseaux neuronaux profonds pour des tâches telle que la classification d'images ou de textes. Ces méthodes sont principalement fondées sur la visualisation des *imputs* - ici, la matière textuelle - prises en compte par le réseau pour construire une décision. Cependant, ces techniques que l'on retrouve dans LIME, SHAP, Grad-CAM ou TDS, nécessitent un effort supplémentaire pour interpréter la visualisation. Dans cet article, nous proposons une nouvelle approche pour inspecter les couches cachées d'un réseau CNN ajusté afin d'extraire des objets linguistiques directement interprétables des textes. En particulier, nous détaillons une extension pondérée de la mesure *Text Deconvolution Saliency* (wTDS). Nous démontrons empiriquement l'efficacité de notre approche sur des corpus de deux langues différentes : anglais et français. Sur tous les jeux de données, wTDS encode automatiquement des objets linguistiques complexes ou motifs basés sur les co-occurrences et éventuellement sur l'analyse grammaticale et syntaxique.

INDEX

Mots-clés: deep learning, apprentissage profond, convolution, déconvolution, texte, cooccurrence, motifs
Keywords: deep learning, convolution, deconvolution, texte, cooccurrence

AUTHORS

LAURENT VANNI

Univ. Côte d'Azur, BCL, UMR UNS-CNRS 7320, Nice, France

MARCO CORNELI

Inria, CNRS, Laboratoire J.A. Dieudonné, Maasai research team, Nice, France
Univ. Côte d'Azur, Center of Modeling, Simulation and Interactions, Nice, France

DAMON MAYAFFRE

Univ. Côte d'Azur, BCL, UMR UNS-CNRS 7320, Nice, France

FRÉDÉRIC PRECIOSO

Inria, CNRS, Laboratoire J.A. Dieudonné, Maasai research team, Nice, France
Univ. Côte d'Azur, I3S, UMR UNS-CNRS 7271, Nice, France