Deconvolution for linguistic analysis

L. Vanni[1], V. Elango[2], C. Aguilar[1], D. Longrée[3], D. Mayaffre[1], F. Precioso[2], M. Ducoffe[2]
[1] Univ. Nice Sophia Antipolis - I3S, UMR UNS-CNRS 7271 06900 Sophia Antipolis, France
{lvanni, mayaffre}@unice.fr
[2] Univ. Nice Sophia Antipolis - BCL, UMR UNS-CNRS 7320 - 06357 Nice CEDEX 4, France
{ducoffe, precioso}@unice.fr - ecveer@gmail.com
[3] Univ. Liège - L.A.S.L.A, Bélgique
dominique.longree@uliege.be

Abstract

This document contains the instructions for preparing a paper submitted to COLING-2018 or accepted for publication in its proceedings. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut nec tellus at lectus suscipit porta. Sed ut aliquam tellus. Ut in arcu nec dui tincidunt suscipit in ultrices sem. Curabitur sed nibh quis est tincidunt aliquet. Nullam et nulla lorem. In hac habitasse platea dictumst. Nam aliquam nisi vel orci venenatis venenatis. Nunc aliquet nibh ut odio dapibus, sit amet convallis quam tincidunt. Suspendisse potenti. Vestibulum ut erat ac mauris imperdiet venenatis. Aenean porttitor mollis mi, eu placerat enim bibendum sodales. In vel ligula diam. Aenean faucibus lacinia rutrum. Aenean in risus neque.

2 Related work

Mauris ut magna ut diam hendrerit tincidunt. Duis turpis lacus, lacinia ut accumsan a, rutrum eu justo. Donec efficitur purus non leo iaculis elementum. Nulla pulvinar ligula ut pretium vulputate. Mauris non suscipit felis, ac molestie sem. Duis quis lacus sed massa pharetra eleifend non ut urna. Phasellus lobortis mattis pharetra. Phasellus mattis purus non quam molestie tincidunt.

Sed hendrerit at leo sed tristique. Vestibulum fringilla, nisi id rutrum congue, erat elit hendrerit mi, eget tempus erat nisl sit amet urna. Pellentesque ornare, nunc vel molestie scelerisque, ante augue condimentum orci, sollicitudin sodales odio neque at tortor. Mauris pellentesque ex neque, ut finibus ante pulvinar nec. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus et fermentum urna. Donec rutrum, ex vel interdum mattis, nibh augue mattis magna, vel varius ante ipsum sed diam. Sed dui risus, gravida sit amet ornare nec, lacinia id elit. Nunc consectetur commodo ante semper suscipit. Quisque ullamcorper mauris id arcu placerat pellentesque. Sed sit amet dolor metus.

3 Model

3.1 Text Classification

We propose a deep neural model to capture linguistics partern of the text. This model is based on simple Convolutional Neural Network with an embedding layer for words representaion, one covolutional with pooling layer and finnaly one Dense layer. Figure 1 shows the global structure of our architecture. The input is a sequence of words $w_1, w_2 ... w_n$ and the output contains class element (for text classification). The embedding is build on top of a Word2Vec architecture trained on a Skip-gram model. Our text tokenizer keeps all the words to make sure all linguistics material could be detected at the end by the model. This embedding is also trainable by the model to reach the best text-classification acuracy.

The Convolutional layer is based on a 2 dimensionals convolution, the same as used for pictures convolution, but with a fixed width corresponding to the max width (this size is actually equal to the embedding size). With this setting, our usage of the 2 dimensionals convolution is in reallity the same as a 1 dimensional convolution (the default convolutionnal layer for text). The only parameter we adjuste here is the height of the filter corresponding to the number of words we want to put in the filter. The goal of this approach is to be able to use the standard picture deconvolution (conv2D Transpose) methods for our model on text.

The last layer is a fully connected dense network (with one hidden layer) finishing on a output size corresponding to the number of class we attempt to train.

3.2 Deconvolution

Since we use same architecture as image detection, making a deconvolutional layer is really straightforward. There are several methods to visualize the deep internal mecanisms of a neural network. One is called convolutional transposed. Our deconvolutional network use the same embedding and convolution layer as we use for the classification but we replace the finale dense layer by a convolutional transposed layer (also called deconvolution). After we trained the model we setup the weight of each neuron of the deconvolutional network with the learned weights of the classification network. The result is a new network that takes in input a sequence of words and gives us in output all the trained filters of the text classification applied on the given sequence. Then the activation score of each word is calculated as shown in Equation 1 with x is the size of the embedding, and y the number of applied filters :

$$\sum_{i=1}^{x}\sum_{j=1}^{y} a_{ij} = s_w \tag{1}$$

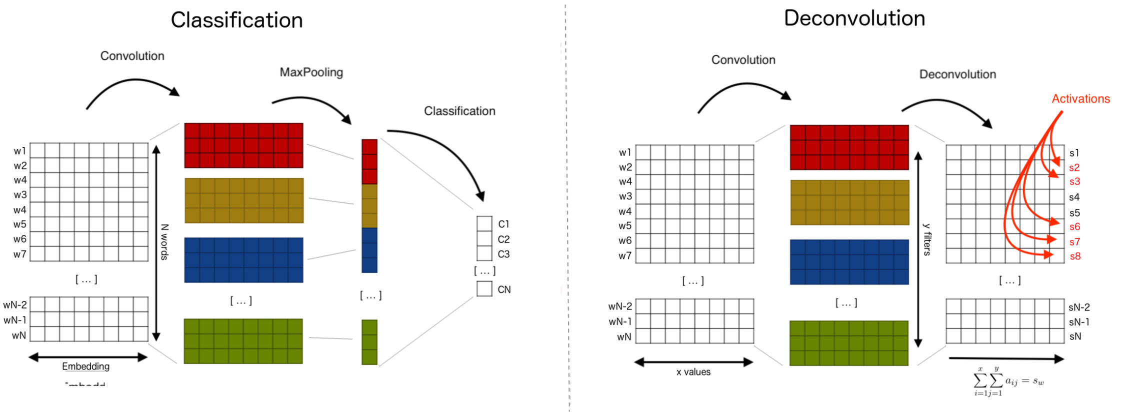

Figure 1: Deconvolution model

With this method we are able to show a sort of topology of a sequence of words. All words have an unique activation score related to the others. We going to see now that this output of the deconvolution give us many information on how the network takes his final descision (prediction). There're well known linguistics marks encoded inside but also some more complexe pattern based on cooccurrences and maybe also on grammatical and syntaxic analysis.

4 Experiments

4.1 Z-score Versus Activation-score

Z-score is one of the most used methods in linguistic statistics. It compares the observed frequency of a word with the frequency expected in case of a "normal" distribution. This calcul gives easily for example the most specific vocabulary of a given author in a contrastive corpora. The highest z-score are the most specific word in this case. This is a simple but strong method to analyze feature on text. It can be also used to classify word sequences according to the global z-score (sum of the score) in the sequence. The mean accuracy of this methods on our data set is around 85%, that confirm z-score is really meaningful on contrastive data. On the other hand, the deep learning reaches most of time more than 90% on text classification. It means the training methods can learn also by themselves some sort of linguistic specificities useful to distinguish class of text or authors. We've seen on image that's the role of the convolution. It learns an abstraction on the data to make classification easier. The question is : what is the nature of this abstraction on text ? We going to seen now that the deep learning detect automatically words with hight z-score but apparently it's not this only linguistic marks detected.

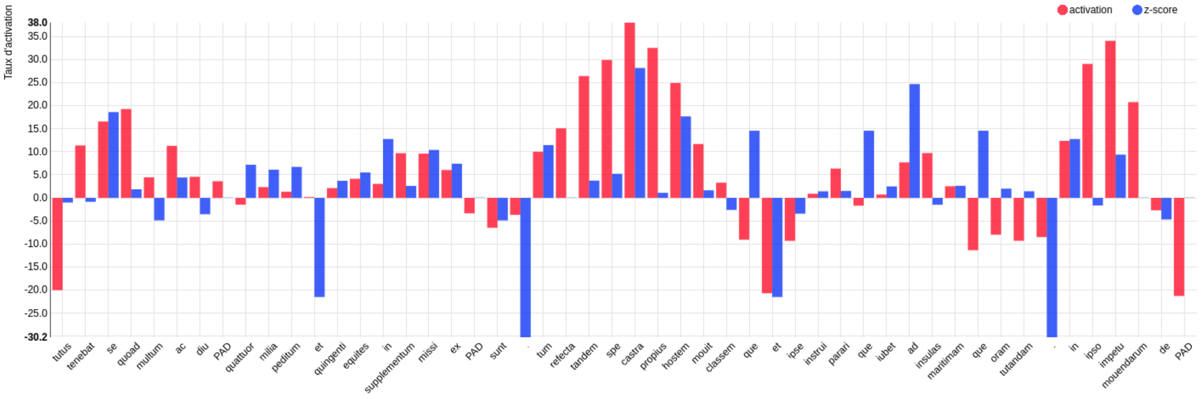

Figure 2: Latin dataset : Livius Book XXIII Chap. 26 - Z-score Vs Activation-score

The Figure 2 shows us a comparison between z-score and activation-score on a sequence extract form our latin corpora. Here it's an example where Livius[1] use some specific words. As we can see, when the z-score is the highest the activation-score around is also very high (word *castra*). But not always, for example small words as *que*, *ad* and *et* are also high in z-score but they not activate the network as the same level. We saw in (reference ****) that deeplearning is more sensible with long words, but we can see also on Figure 2 that word like *tenebat*, *multum* or *propius* are totally uncorrelated. If we make the Pearson[2] correlation coefficient on this sequence, we obtain 0.38. Not a hight correlation so, and this example one of the most correlated example of our dataset.

4.2 Dataset : English

4.3 Dataset : French

" ...notre pays advienne, à l'école pour nos enfants, au travail pour l'ensemble de nos concitoyens, pour le climat, pour le quotidien de chacune et chacun d' entre vous. Ces transformations profondes ont commencé et se poursuivront avec la même force, le même rythme, la même intensité"

Cet extrait issu du discours de vœux d'Emmanuel Macron (31 décembre 2017) est mal attribué par l'ADT (Z-score) qui le rapproche statistiquement de De Gaulle, et bien attribué par le Deep learning qui reconnait du Macron. L'erreur d'attribution statistique s'explique par une phraséologie très gaullienne et la multiplication de marqueurs linguistiques fortement indicés chez de Gaulle : par exemple, de Gaulle avait pour caractéristique de faire des phrases longues et littéraires articulées autour de conjonctions de coordination comme " et " (z-score = 28 chez de Gaulle, 2 occurrences dans l'extrait). Son discours était également plus conceptuel que la moyenne, et cela se traduisait par une sur-utilisation des articles définis (le, la, l', les) très nombreux dans l'extrait (7 occurrences) ; particulièrement au féminin singulier (" la " République, " la " liberté, " la " nation ", " la " guerre, etc ; ici " la " même force, " la " même intensité).

Les meilleures performances du deep learning interrogent quant à elle le linguiste et épouse parfaitement ce que l'on sait socio-linguistiquement du discours dynamique de Macron. La zone d'activation la plus importante de l'extrait concerne le syntagme nominal " transformation profonde ". Pris séparément,

[1] Lucius Livius Andronicus (c. 284 – c. 205 BC) was a Greco-Roman dramatist and epic poet of the Old Latin period

[2] Pearson correlation coefficient measures the linear relationship between two datasets. It has a value between $+1$ and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative

aucun des deux mots du syntagme n'est très macronien d'un point de vue statistique (" transformation " = XXX ; " profonde " = YYY). Mieux : le syntagme lui-même n'est pas attesté dans le corpus d'apprentissage du Président (0 occurrence). Cependant, on peut constater que la co-occurrence de " transformation " et de " profonde " s'élève à +XXX chez Macron : ce n'est donc pas l'occurrence d'un mot seul, ou de l'autre, qui est macronienne mais l'apparition simultanée des deux dans une même fenêtre. Pour autant, la cooccurrence de " transformation " et de " profonde " ne saurait être suffisante pour caractériser Macron, notamment parce que la cooccurrence des deux mots est plus fréquente encore chez Pompidou par exemple ; d'autres indices additionnés sont nécessaires à l'attribution. Les zones d'activation secondes et complémentaires de l'extrait concernent ainsi les deux verbes " advienne " et " poursuivront ". D'un point de vue sémantique, les deux verbes conspirent parfaitement, après le syntagme " transformation profonde ", à donner la dynamique nécessaire à un discours qui prône le changement. Mais c'est les temps verbaux (portés par la morphologie des verbes) qui apparaissent déterminants dans l'analyse. Le calcul des codes grammaticaux co-occurrents au mot " transformation " indique ainsi que les verbes au subjonctif et les verbes au futur (et également les noms) sont les codes privilégies chez Macron. (GRAPH XXX) Plus précisément encore, l'algorithme indique que, chez Macron, lorsque " transformation " est associée à un verbe au subjonctif (ici "advienne "), alors il existe le plus souvent un verbe au futur co-présent (ici " poursuivront "). " Transformation profonde ", " advenir " au subjonctif, " poursuivre " au futur : tous ces éléments signent, ensemble, un discours fait de promesse d'action, dans la bouche d'un président jeune et dynamique. Enfin, le graphique indique que " transformation " est surtout associée aux noms chez le Président : dans une concentration extraordinaire, l'extrait en recense ainsi 11 (" pays ", " école ", " enfants ", " travail ", " concitoyens ", " climat ", " quotidien ", " transformations ", " force ", " rythme ", " intensité ").

4.4 Dataset : Latin

5 Conclusion

Mauris ut magna ut diam hendrerit tincidunt. Duis turpis lacus, lacinia ut accumsan a, rutrum eu justo. Donec efficitur purus non leo iaculis elementum. Nulla pulvinar ligula ut pretium vulputate. Mauris non suscipit felis, ac molestie sem. Duis quis lacus sed massa pharetra eleifend non ut urna. Phasellus lobortis mattis pharetra. Phasellus mattis purus non quam molestie tincidunt.

Sed hendrerit at leo sed tristique. Vestibulum fringilla, nisi id rutrum congue, erat elit hendrerit mi, eget tempus erat nisl sit amet urna. Pellentesque ornare, nunc vel molestie scelerisque, ante augue condimentum orci, sollicitudin sodales odio neque at tortor. Mauris pellentesque ex neque, ut finibus ante pulvinar nec. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus et fermentum urna. Donec rutrum, ex vel interdum mattis, nibh augue mattis magna, vel varius ante ipsum sed diam. Sed dui risus, gravida sit amet ornare nec, lacinia id elit. Nunc consectetur commodo ante semper suscipit. Quisque ullamcorper mauris id arcu placerat pellentesque. Sed sit amet dolor metus.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.