AIX-MARSEILLE UNIVERSITY
M2 IAAA

Are there some languages more difficult to analyse ?

ZHOU Baicong - DEBERNARDI Hippolyte - FERSULA Jeremy

December 22, 2019

Contents

Introduction

The natural language processing is recently becoming a well focused aspect in artificial intelligence. But the natural languages are enough complex sometimes to confuse a experienced researcher in this aspect.
The objective of this project is to analyse the reason why the same analyser performs differently in the same situation on different languages. We can see this phenomenon in Figure 0.1. What we are going to do is analyse the reasons using the analyser MACAON and try to understand the hiding factors behind the differences of the performance judged by LAS (Labeled Accuracy Score) and UAS (Unlabeled Accuracy Score). In the table, we can see that the performance of MACAON on different languages are very different. Some of them are very high, for example: Hindi has a LAS of 79.47% and a UAS of 86.80% while the Turc scored 47.28% on LAS and 55.20% on UAS. Our main purpose is to explain this phenomenon on the given data set, which has around 20000 words for each language.

L	LAS	UAS	L	LAS	UAS	L	LAS	UAS
hi	79.47	86.80	ru	69.70	73.85	sl	63.47	71.78
it	78.38	82.15	da	68.12	74.18	hr	63.58	72.10
ur	76.33	83.55	id	67.05	72.21	cs	63.84	72.45
pl	76.18	84.41	en	67.18	74.39	lv	62.30	69.83
ja	75.74	85.60	es	66.93	74.52	hu	62.73	68.86
no	73.25	78.91	uk	65.85	74.19	fi	62.77	70.83
bg	73.40	82.36	ro	65.13	72.53	zh	59.91	65.15
el	72.55	78.52	ga	65.13	74.02	vi	59.77	62.68
ca	72.06	79.70	fa	65.22	73.42	eu	58.80	68.78
sv	71.10	77.36	he	64.68	72.34	nl	57.44	68.43
fr	71.36	77.02	et	64.76	75.40	ko	53.12	63.21
pt	70.73	76.95	ar	64.28	71.65	tr	47.28	55.20

Figure 0.1: This figure shows the performance of this analyser on 36 different languages in the same situation. LAS stands for Labeled accuracy score and UAS stands for Unlabeled accuracy score

1 Presentation of MACAON

MACAON is a set of tools designed for dealing with ambiguous input and extending the inference of input module in a global scope. It consists of several modules, which can perform classic NLP(Natural language processing) tasks, for example: tokenization, word recognition, part of speech tagging, word formation, morphological analysis, partial or full parsing for native text or word case.
Generally speaking, the MACAON module can be seen as an annotation device that adds a new level of annotation to its input, usually depends on the previous module's annotation. These modules communicate with XML files that allow different levels of annotations to be represented as well as ambiguity at each level. In addition, the processing phase maintains the original XML structure of the unprocessed file (the logical structure of the document, information from the automatic speech recognition module...).
As mentioned before, one of the main features of MACAON is that each module has the ability to accept ambiguous input and generate ambiguous output, so that ambiguity can be resolved in the subsequent processing stage. The compact representation of the ambiguous structure is at the heart of the MACAON interchange format, as described in Section 2. In addition, each module can weigh the solutions it produces. Such weights can be used to rank or limit the number of solutions for subsequent processing. The main difference between MACAON and these methods is that MACAON defines the exchange format between NLP modules, not the annotation format. More precisely, this format is dedicated to the compact representation of ambiguity: some of the information represented in the exchange format will be interpreted by the MACAON module and will not be part of the annotation format.
Despite the fact that MACAON is a remarkable analyser, we did not use MACAON to process the features that we have formed in this project since we had some difficulties in using it when we were working on the project. We have tried to use this analyser but it is not very easy for us to use. Instead, we used directly the values given in the problem.

2 Methodology

In the processing of the 36 languages mentioned in the chart, we tried to give each language a vector of features in the following form:

$$x_i = \begin{pmatrix} f_{i1} \\ \vdots \\ f_{in} \end{pmatrix} \tag{2.1}$$

In these vectors, each element refers to a feature, such as average distance to the root of the sentence, average length of the sentence or even the mean length of successive dependency. So f_{ij} means the j-th feature in the i-th language and x_i means the vector of features of the i-th language. To do the regression, we use the values mentioned above, LAS and UAS. We also form a vector for them to do the multiple linear regression:

$$Y_i = \begin{pmatrix} y_{i1} \\ y_{in} \end{pmatrix} \tag{2.2}$$

where Y_i is the value vector of i-th language and y_{i1} and y_{i2} are the LAS and UAS of i-th language respectively.
To form our model, we do four types of regression: linear regression, ridge regression, Lasso regression, elastic net regression. We can get a matrix after the regressions are done. we note the matrix we get as W. And we will add subscript to W when it comes to specific methods. With these matrix, we have:

$$\widehat{Y} = WX \tag{2.3}$$

and $\widehat{Y}$ means the predictions of our models generated by the regressions. Also, we will add subscripts to $\widehat{Y}$ when we refer to specific methods.
To evaluate our models, we use the R^2 coefficient. The R^2 gives the proportion of variability of y which is explained by the model. The closer the R^2 is to 1, the better the model fits the data. By definition, we have:

$$R^2 = \frac{SCE}{SCT} \tag{2.4}$$

In this formula, SCE means variability explained by the model(Somme des Carres Expliques), and SCT means total variability of the endogenous variables (Somme des CarrÃ©s Totaux). By the definition of SCE in SCT, we have:

$$SCE = \sum_{i=1}^{n} \left(\hat{y}_i - \bar{y}_n\right)^2 \tag{2.5}$$

and

$$SCT = \sum_{i=1}^{n} \left(y_i - \bar{y}_n\right)^2 \tag{2.6}$$

In these formula $\hat{y}_i$ refers to the i-th predicted values of the labels given by our model. And y_i is the the actual values of the labels. $\bar{y}_n$ is the average value of the actual label.

3 Features and results

3.1 Set of features and preliminary results

We tested a total of 28 different features, including the mean proportion of any UPOS tag in a sentence (accounting for 18 of them, since we also counted the missing tags as a feature). In order to have a basic idea on the most important features we tested, we calculated a pearson correlation coefficient for each individual feature and plotted the results of the feature over the score.

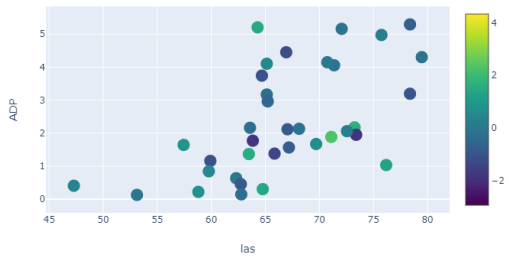

Figure 3.1: ADP / las (r=0.58)

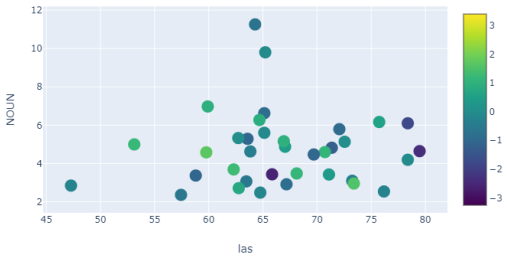

Figure 3.2: NOUN / las (r=0.04)

Regarding the features we "built" ourselves, we tried to think of ways to measure the complexity of the data for each language. As an example, we measured the mean length of sentences, thinking that it may be more difficult for an analyser to parse and label accurately long sentences. We also tried several more complex measures, such as what we call the max nested dependency, which is the maximum number of steps we must go through going from word to head until we reach the root of the sentence.

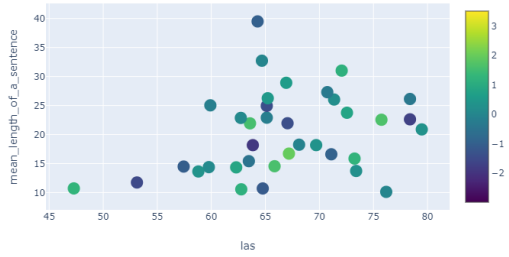

Figure 3.3: Mean Sentence Length (r=0.28)

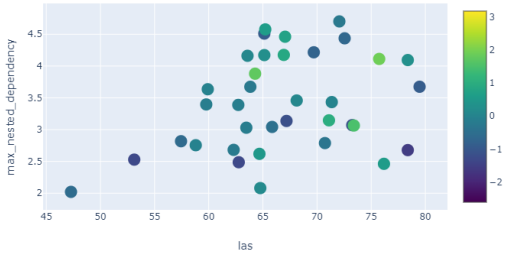

Figure 3.4: Max Nested Dependency (r=0.32)

In order to know which feature is really relevant, we decided to establish a correlation table between all of our variables :

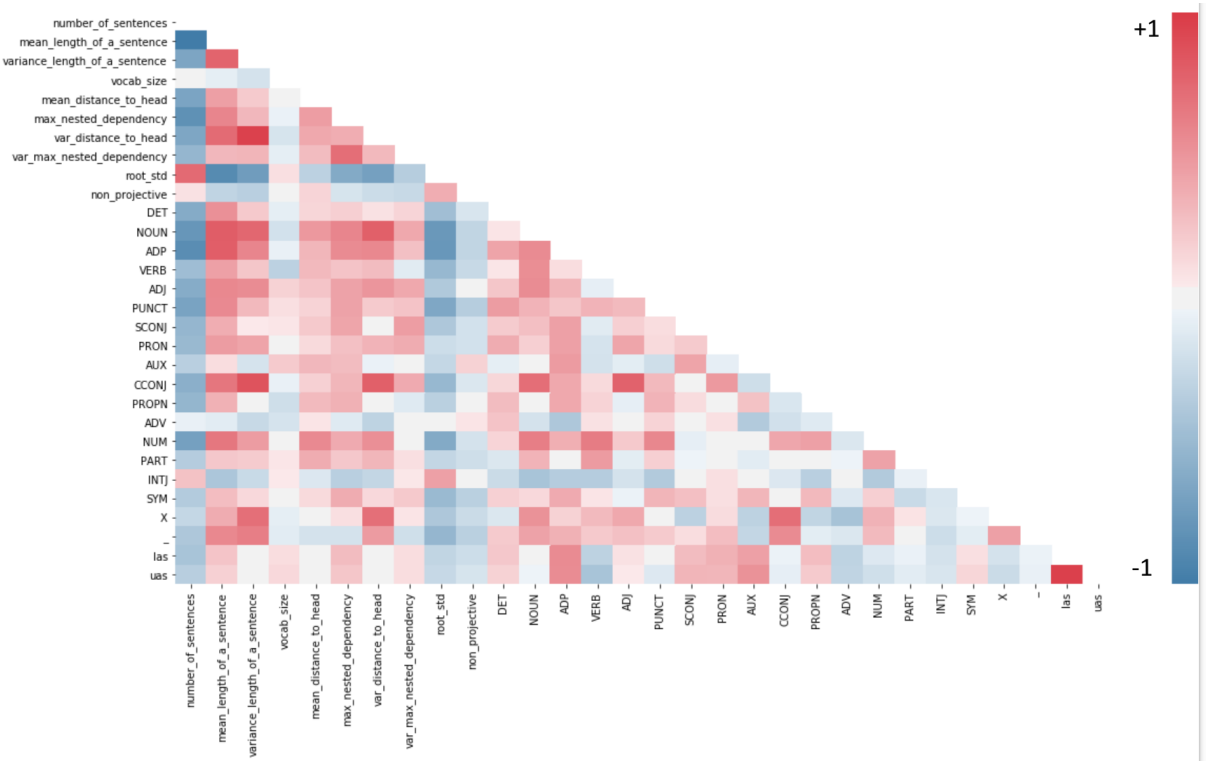

Figure 3.5: Pearson correlation coefficients, all variables crossed

This table shows us a lot of information, some of which are quite predictable. As an example, we can see that the longer sentences are on average, the less the proportion of each UPOS tag per sentence. Something important that we can notice on this table is the lack of important correlation between features that may predict the scores. We see that the max nested dependency is not strongly correlated to the mean sentence length and ADP proportion, and if these features explain well the variance of the scores we already know that they are quite independant.

3.2 Performances and important features

Considering all of our features together, we performed a simple linear regression over all of the data. We have the following results : For both the scores, the most important feature by

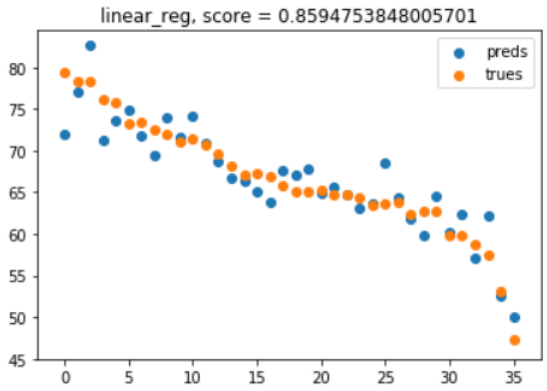

Figure 3.6: Linear regression on LAS
R^2 = 0.86

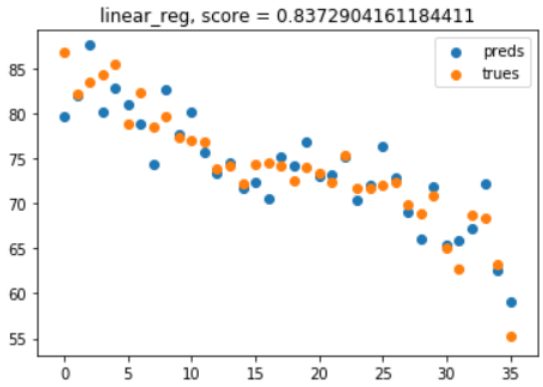

Figure 3.7: Linear regression on UAS
R^2 = 0.84

absolute value of the regression's coefficient is the **proportion of interjections** in the corpus. It has here a coefficient of roughly -159 when every other feature is much closer to 0 (mean coefficient = -1.7). The negative coefficient denotes that the more a corpus has interjections, the lower the score of the analyser is for that corpus. The second place is interesting but will be discussed later.

The third and fourth most important features in LAS according to this regression are the proportion of unknown UPOS tag and and the proportion of non-projective sentences, with a coefficient of -19. Regarding the unknow UPOS tag, this is quite understandable, because the analyser probably did not perform well the labeling if it labeled words as unknown.
The proportion of non-projective sentences is quite interesting, because we know *a priori* that transition based parsing methods works better on projective sentences. We find this feature as an important feature in UAS scoring as well, but in 6th position (behind numerals and particles).

Regarding the second place, it is held by the proportion of symbols in the corpus, with a coefficient of roughly +28. The fact that this coefficient is positive is quite strange, and is probably the cause of a few outliers that happens to have both a lot of symbols and a good score. In fact, most of the languages have close to no symbols at all in their corpus. Here is a relevant plot :

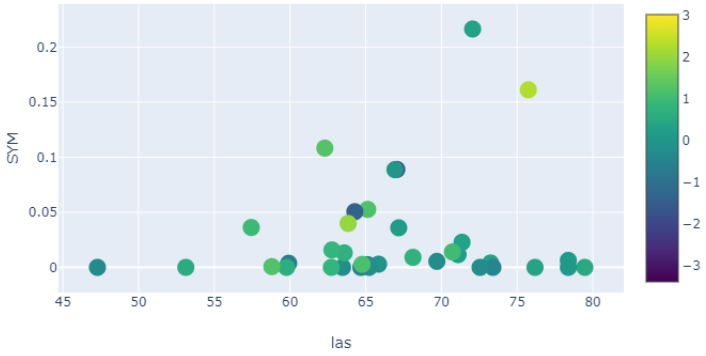

Figure 3.8: SYM / las

We can see on this plot two points in the upper right corner. The upmost point is catalan, the other is japanese. We clearly see that they both have a good score and a lot of symbols in their corpus, and are distant from the other points.

In order to have more visual evidences for the correlation of the good features, and the absence of outliers, we plotted their regression line :

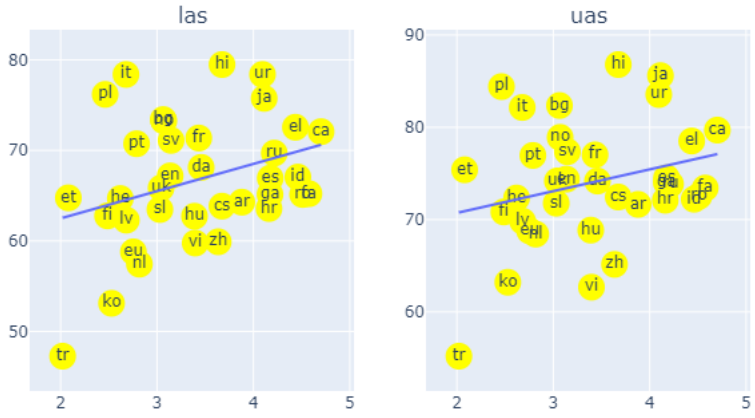

Figure 3.9: Max nested dependency regressions

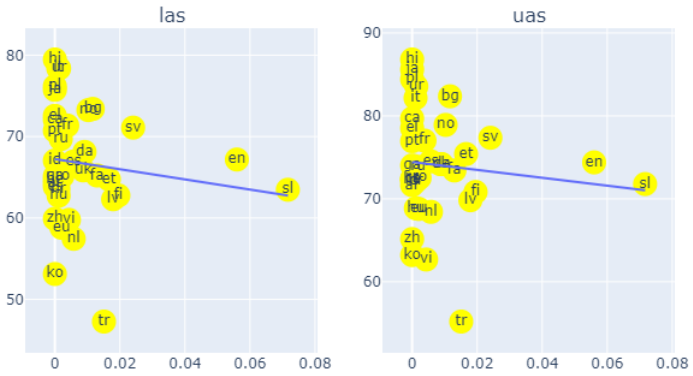

Figure 3.10: INTJ regressions

Here, we see something quite important, as the nested dependency line fits well all of the languages, the **INTJ seems to rely on outliers**. Unfortunately, this problem showed up too late in the project in order to be tackled correctly. Something that could have been done is simply re-doing a regression without the features INTJ and SYM.

3.3 Alternative methods

We tried to perform several different methods of linear regression over the data. Here are as an example the results of the Ridge regression :

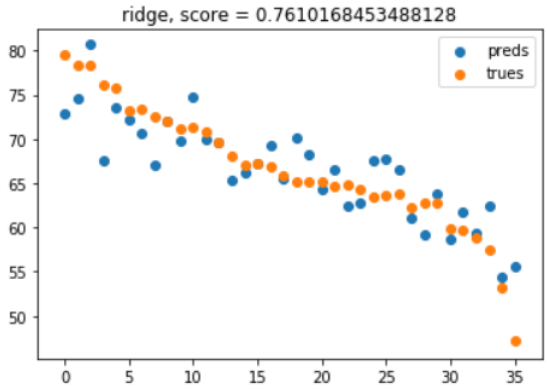

Figure 3.11: Ridge regression on LAS
R^2 = 0.76

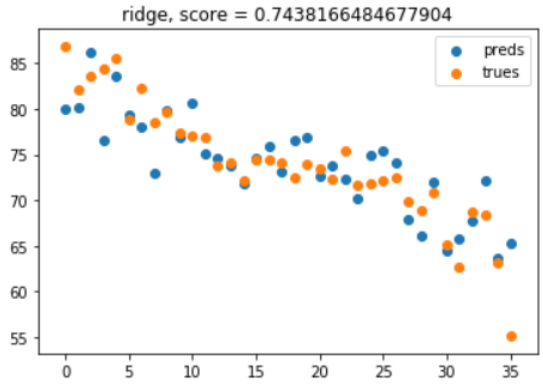

Figure 3.12: Ridge regression on UAS
R^2 = 0.74

However, at the end of the project, we began to question the importance of regularization in such a problem. We decide in the end not to take in account the results we obtained on Ridge, Lasso, and Elastic-net. The main argument that motivates such a decision is the fact that we are not trying to make a good model in generalisation. We need in fact to explain a finite data set and the reason there is disparity in this data set in particular. We don't have a clear and simple way to differentiate which part of our results is due to problems inherent to the languages or only problems inherent to the data. In order to overcome such an ambiguity, we would need very large and diverse data, and we would need to have some certainties relatives to the production of the texts. The peculiar thing here is that we could even observe significant differences between two texts that share the same topic, only because their authors are different.

Something that could have been done to answer this problem would have been to split the texts directly. Then, we could have had enough similarity between train and test to learn the real impact of the features we chose on the scores.

4 Conclusion

In the end, our linear regression explain up to 86% the variance of the LAS score and 84% the variance of the UAS, considering the R^2 as a percentage. This is relative to our data, and not the nature of the languages, and we have features that rely on outliers. Overall, the main relevant thing that we can conclude is that the maximum nested dependency plays a strong role in the variations of scores in the data set.

We have done the processing of 36 natural languages and analysed why the performances of the same analyser are different on these 36 languages. But there are still several missing points in our works. First, the performance of the processing highly depends on the feature engineering. In our work, we haven't put too much features for fear of the over-fitting and a high time consumption of the program. But with a stronger GPU and a well designed feature engineering, we may be able to improve our results and have a better performances.

Second, we can try to use MACAON to actually process these languages and see what is going to happen. We have achieve that in this project since it is a bit difficult for us to use, but we may also learn something by actually achieving it.

Third, we have constructed our model on the 36 languages that are given in the project. But there is 4000-8000 languages on earth, some of them are extremely rare and very different from the well known languages. The performance of our model and the MACAON on other languages are still unknown. We may have the chance to test our model and MACAON on these languages to improve our work.

Forth, we have solve the main problem but we have not got any chance to do some further study. In the future, we may be able to use MACAON in other circumstances, by that time, we may be able to use it well in the study of the projects of natural language processing.

Appendices

Built features

Here is a list of features we built and their associated pearson coefficient for LAS :

- Number of sentences in the corpus **0.37**
- Mean length of sentences **0.28**
- Variance of length in the corpus **0.07**
- Vocabulary size **0.15**
- Mean distance to head **0.02**
- Variance of the distance to head **0.32**
- Max nested dependency **0.04**
- Variance of the max nested dependency **0.15**
- Standard deviation of the position of the root **0.23**
- Proportion of non-projectives sentences in the corpus. **0.18**