Universitat Politècnica de Catalunya

Master in Artificial Intelligence

Introduction to Natural Language Processing

Classification of Documents about Organizations

Authors:
Miquel Perelló Nieto
Marc Albert Garcia Gonzalo

Date:
December 16, 2012

Contents

1 Introduction

This document describes how we have implemented a system for classifying documents regarding different kinds of organizations. Specifically *"team"*, *"company"*, *"university"* and *"other organizations"*.

The system uses modern Machine Learning techniques for the classification algorithm. A model selection process is followed to decide which of the different models, and with which parameters, is the one that classifies better, without overfitting the training data.

We mainly focus on the aspects of the process related to Natural Language Processing, while being correct in all the aspects regarding more general Machine Learning procedures. The main aspects we have focused on are the feature extraction, and the different kinds of classifiers which could be used.

The program used for performing all the experiments has been developed from scratch using the Python programming language. For the classification algorithms, we have used an available library.

2 Data set

The data set for this project comes in a Python pickle structure, containing information gathered from Wikipedia. The information is extracted from regular Wikipedia pages, representing different types of organizations.

These organizations, and for the porpouse of the classifier, are grouped in next categories:

- Companies
- Universities
- Sport teams
- Other organizations

In the pickle file we have all the relevant information for each organization from Wikipedia. This information includes the name, alternate names, information from infoboxes, and sentences on the body of the page, containing the name or an alternate name of the entity.

Labels come in separate files, and are available only for companies, universities, and sport teams. This is important, as we do not have a specific label for other organizations, which means that we are using as other organizations any which is not labeled, not the ones which do not belong to the three other categories. This means that there are companies, universities, and sport teams which we will label as other organizations. Therefore this will create noise in the dataset, and it will be harder for the classifier to find patterns that exist in one category, and not the others, as some items will be mixed.

The number of instances in each of the group is next:

Companies 1293 (60.85%)

Universities 289 (13.60%)

Sport teams 43 (2.02%)

Other organizations 500 (23.53%)

Note that the number of other organizations has been selected by us, approximating it to a quarter of the instances, because we have four classes. Some other criteria could be used.

3 Technical solution

The implementation of the whole system has been developed in Python (version 2.7).

Next libraries have been used:

numpy for array manipulation, and linear algebra operations

scikit-learn for machine learning algorithms

Next there are explained the files used in our project

run.py main project file, it iterates over the different parameters of the learning algorithms, and calls all process functions

parse_input.py loads data from pickle file, and labels entities with information from category files

divide_dataset.py splits dataset in a part for training, and another part for validation

feature_extraction.py extracts features from the original dataset. Extracted features are mostly based on the number of words existing in certain groups. Exact features are described in the section *Feature extraction*

feature_extraction_histogram.py extracts features for the adaptive chi squared kernel. For this kernel, instead of using groups of words, we have used a histogram with the number of times certain words appear

classifier.py trains the different models, and calculates the accuracy obtained with validation data

top_words.py functions to extract the most common words of each class, avoiding English stop words, and based on some parameters

set_words.py functions to extract different subsets of words; like union, intersection or substraction; for classes, avoiding English stop words.

counter.py backport of the Counter class from Python 2.7, to be used in Python 2.6 or earlier. This code is not original.

4 Feature extraction

In this section we will define which features have been extracted to train our models. We will see some features that have been used for decision trees and Support Vector Machine with radial kernel. And finally the Histogram features that corresponds only to the Support Vector Machine with adaptive chi squared kernel.

4.1 Name length

First feature we exctracted was the length of the name of the organization, as we have detected some correlation between this length, and the associated class.

In the figure 1 we can see that university names are usually larger than company names , and in the figure 2 there are the two features (length in characters and in words).

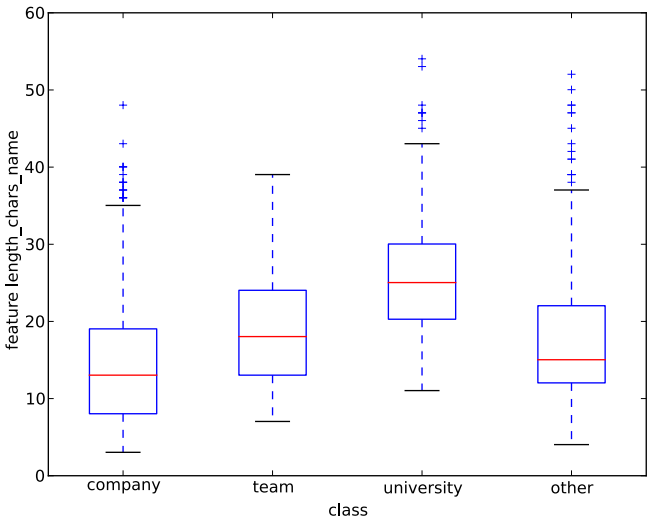


Figure 1: Feature: number of characters in the name.

4.2 Top words

Then we started thinking about which features could give us more information, and we thought that we could select the top words in the names and in the sentences for each kind of organization.

First, we have removed a list of English stopwords because in our opinion are noise for our purpose. These words are very common and do not give any information about the class of the organization; some examples of these words are : 'a', 'an', 'the', 'is', 'one'.

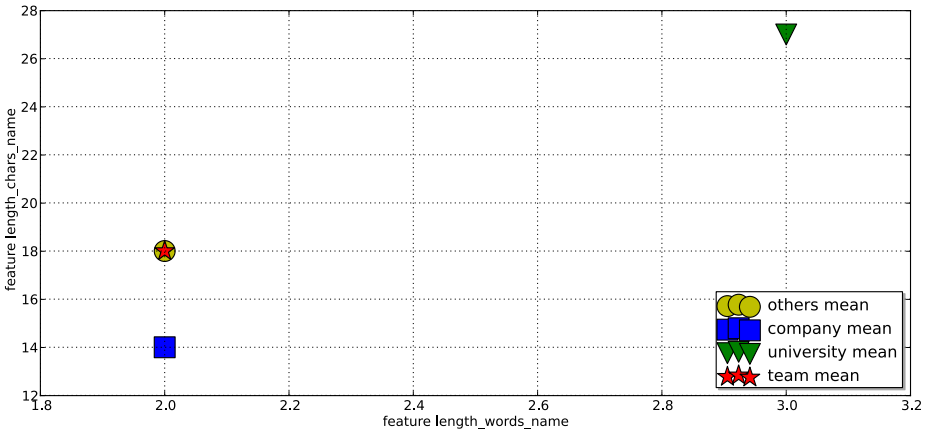
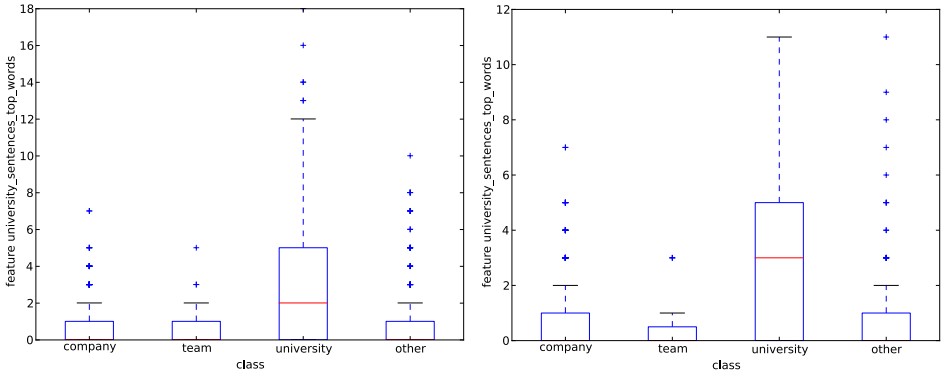


Figure 2: Features: length of the name in characters vs words.

Next, we created eight new features, two with the number of top words for each kind of organization, one for the words appearing in the name, and another for the words appearing on the body of the text (the sentences variable). The figure 3 is an example of one of them, we see how it exists a correlation between these variables and the kind of organization.



(a) In the training set.

(b) In the test set.

Figure 3: Feature: number of coincidences in the top words of universities.

4.3 Subset words

Besides these eight features, we wanted to find more. One new idea was to create different subsets of words. The idea is represented in the figure 4. The features are the different intersections and the unique words per organization. In the picture we see a simplified representation with only three of the organizations, but in the problem there are four.

Then the different subsets are :

- Four subsets with the words that only corresponds to one organization (example : with team is : $T - C \cup U \cup O$).
- Six subsets with the intersection for each pair of organizations and removing intersections with other organizations. The number of features of this kind is all the pairs with four elements $\binom{n}{r} = \binom{4}{2} = \frac{4!}{2!2!} = 6$ (example : with team and company is : $T \cap C - (O \cup U)$).

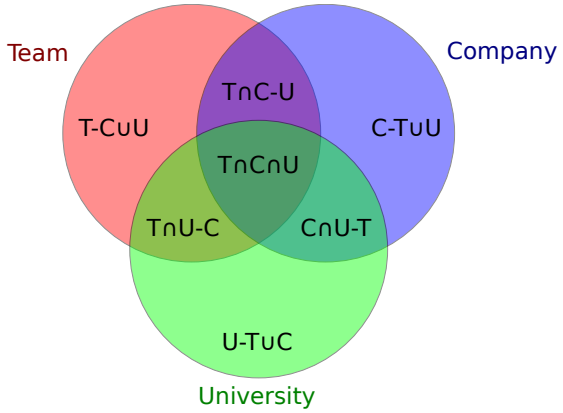


Figure 4: Training words for each category

Here it is an example of the results obtained in the training set, specifically, the one for the unique words in the class sport teams: leones, zebras, osakas, 81-82, brieg, allstar, anorthosis, 695.

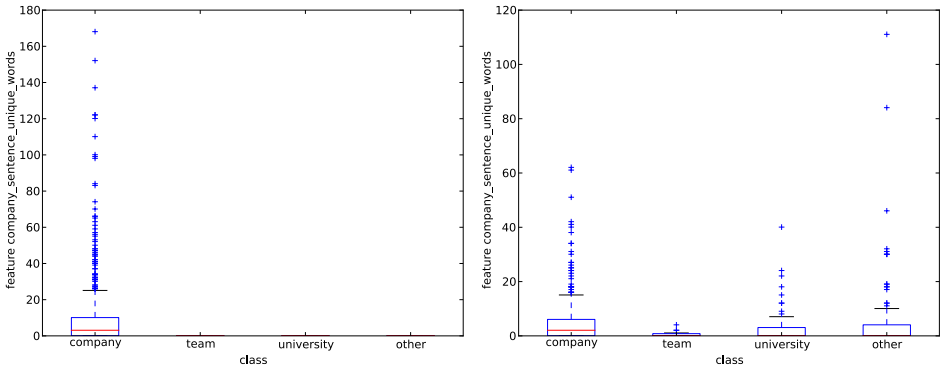
Another example for unique words in universities: accountancy, adequately, advancement, 5700, bachelor, co-education, coursework, debaters, gymkhana, homework, http//www919thebuzzcom/, icici, judo.

We can see that some of the words can be useful for discrimination, but there are a lot that does not give any information and can be considered noise. We could expect that the set of top words would improve its discrimination, as a function of the size of the dataset. With a much larger dataset, we would expect to have more discrimination power in these features.

We created one feature per intersection. The value of the features is the number of top words in the intersection, which are present in the document. These features looks to be quite representatives of the kind of organization, as seen in the figure 5.

4.4 Histogram

In this case the features are extracted as a histogram with one word per feature. Each feature will be the number of occurrences in the sentences, normalizing, so the histogram sum to one.



(a) In the training set.

(b) In the test set.

Figure 5: Feature: number of words in the unique subset of companies.

The number of words that will form the histogram is variable. We first have extracted all the words in the training set. We removed the stopwords and then we sorted the words by frequency. The words we want for the features are in the middle. This is because the most common words can be in any text, and the last words are very rare (In the figure 6 we can see an example of this selection).

But the hard part is to select the good percentages of words. For that reason these are parameters that we have selected by validation.

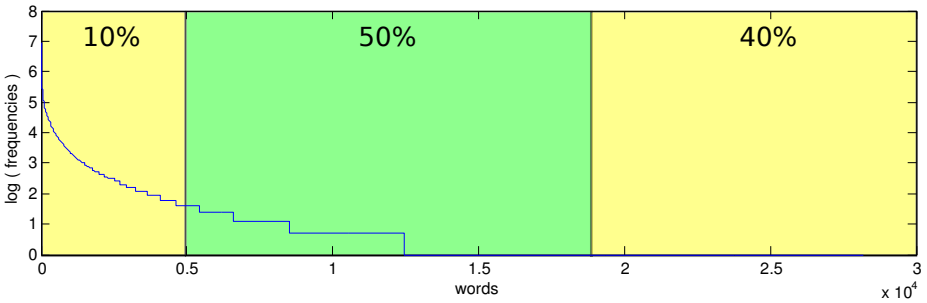


Figure 6: Example of histogram, first 10% and last 40% of the words will be removed; the remaining 50% will be selected as features.

5 Model selection

For choosing the best method for organization document classification, we have trained models of different types, with a set of different parameters, and we checked the accuracy of each model with independent data.

Next there is a short description of the algorithms we used, and the reasoning for choosing them.

Decision tree While being a powerful classifier, decision tree is in general much more interpretable than other algorithms, and we used it to be able to undestand how classification was done by the algorithm

SVM with radial kernel Support Vector Machines are probably the most powerful classification method, being the RBF (radial) kernel the one usually giving better results

SVM with adaptive chi squared kernel This version of Support Vector Machines is design to be used with histagram data, as it could be the number of occurrences of words in documents

For decision trees and SVM with radial kernel, we trained all models using as parameters the number of most common words of each class to be considered. In the case of the SVM with the adaptive chi squared kernel, we used as parameters the percentage of most common words respect to all in the documents to be considered, and a parameter for ignoring a percentage of the most common ones, so we could analyze if most common words are also the most relevant to discriminate.

As this is an academic project focused on Natural Language Processing, we have not optimized other parameters of the classifiers, like the C soft margin parameter of the Support Vector Machines, or the sigma parameter of the radial kernel. It would probably make sense to test the accuracy for different values of these parameters on the best model, to see if the accuracy can be increased.

To select the best model among all the models that we have been trained, we have split the dataset in two different sets, one for training and one for validation. While it is common to use cross validation (e.g. 10-fold cross validation), to have lower bias on the estimated accuracy, we considered that for the scope of this project was not necessary. Our traning dataset was 70% of the available labeled data, and the validation dataset was the other 30%. In case a non-optimistic estimation of the accuracy of the final model could be considered useful, it would be possible to have an extra split on the dataset, and to have an extra dataset for testing the accuracy on the final model. But for this project we focused on analyzing the classification problem from a Natural Language Processing point of view, and not into getting very accurate estimators.

6 Results

In the next table, are shown the results of classifying the validation dataset with different trained models. Each row represents a different value for the parameter which establishes how many of the most common words in all documents of each class is used. This parameter affects directly the feature exactraction process, as there is one feature per class, which contains the number of that set of words present on the document.

The second column contains the accuracy over one obtained using decision trees, and the third column contains the same metric for the Support Vector Machine with radial kernel.

Words used	**Decission tree**	**SVM (RBF)**
1	0.6458	0.6724
3	0.5956	0.6630
5	0.6661	0.6552
8	0.6348	0.6411
10	0.6144	0.6630
12	0.6379	0.6944
20	0.6238	0.6646
50	0.6395	0.6740

Table 1: Accuracy selecting different number of top words.

In this other table it is presented the accuracy over one obtained for classifying the validation data set, with the trained models using different parameters and Support Vector Machine with adaptive chi squared kernel.

In this case, there are two parameters. Each row represents a different value for the percentage of most common words skip in the feature extraction, while in the columns are the percentage considered. As an example, the cell of the table, in the row 20%, in the column 40%, with value 0.6270, means that we used for extracting the features, the 40% of most repeated words for each class, and we discarded the first 20% (over the total) of them. So, we used the words in the range 20% to 40%, and we obtained an accuracy, of 62.70% of documents classified correctly.

	10	**20**	**30**	**40**	**50**
0	0.6082	0.6113	0.6082	0.6050	0.6066
10		0.6003	0.6223	0.6176	0.6207
20			0.6254	0.6270	0.5987
30				0.6207	0.5987
40					0.6097

Table 2: Accuracy selecting different percentages of words; rows are the none considered most common words, and columns corresponds to considered ones.

7 Conclusions

After performing all the experiments described in this document, we have arrived to the conclusions explained in this section.

7.1 Results

The first thing we want to evaluate, is whether the obtained results are good or not. Even before doing the experiments, we could say that the accuracy obtained would be 25% if we assume that we have the same instances for each class, and we classify them randomly. In our dataset, there are around 60% of companies, so implementing a classifier which always predicts company could get a 60% of accuracy on the validation data.

Considering that for most of the models, the classification rate is between 60% and 70%, we believe that the performance of the whole system is very poor.

In next sections, we will try to establish the causes of this bad performance, and suggest possible improvements to the system.

7.2 Dataset

There are two main questions we should answer, regarding the problem data, in order to establish, how it can affect to our results.

- How difficult is for a human to classify entities given our input variables?
- How good is our dataset as a sample of real world data?

The answer to the first question is difficult, and subject to different opinions, but we would say that in general it should be easy for a human to guess whether a text is talking about a university, a company, a sport team, or other kinds of companies. So, in our view, having a text, it should be possible for a machine to classify our domain data, with a very high rate of accuracy. This does not mean that it should be easy to implement this classifier, but rather that with a smart enough classifier the classification is feasible.

Regarding the second question, we do not thing that our dataset is a good sample of real world data. What in our opinion would be the first error, is having the generation of the dataset as a decoupled part of the machine learning process. We believe that better results can be obtained, if it is possible to modify the process of gathering the data, during the training of the models, and after observing the first results. Of course we understand that this is not always possible, depending on the domain.

One thing that does not seem to be considered in the dataset, is the proportion of each kind of organization in real world data. What this does not

have an impact on our validation results (which have been obtained from the same dataset), we assume that our results would be even worse if we could evaluate our models with data not generated by the gathering process of our dataset. What we would to do before gathering any data, is to try to figure out which is the proportion of each kind of organization in all texts about organizations. Then, if we generate our training dataset with this same proportion, we could expect that when using our models in real world cases, they will perform as we expect.

Another consideration, is how reliable is the dataset we are using. Performing a quick view to the entities in the sport team labels, of 43 instances we easily detect three which are actually people, one of a company, and one of a TV show episode. With a more in detail analysis, we could probably find more missclassified entities. A 20% of missclassifications on the dataset would not surprise us. First problem with these missclassifications is related to the instances which will be selected for validation, and that even if they are predicted successfully, they will decrease the classification rates we obtained. But not only. Classification models work because they try to generalize, because they find patterns in data that explain the differences among the different classes we are classifying. With a 20% of missclassifications on our training dataset, we will have the distinctive patterns among classes, repeated in a significant way in all classes, so it will be very hard for the algorithms to find distinctive patterns.

Finally, we mentioned before, that we did not have a specific label for other organizations. So, in addition to the missclassifications, we can expect to have a high number of companies, sport teams and universities in the other organizations class, when training the model. As said in the previous paragraph, this is causing problems for extracting patterns to the algorithm.

So, we can conclude that probably the best way of improving our results, would be to have a more reliable dataset.

7.3 Machine learning algorithms

Comparing the average accuracy of used algorithms, we have next results:

Decision tree 63.22%

SVM (radial) 66.60%

SVM (chi2) 61.20%

As we can see, the algorithm which looks to be performing better is the Support Vector Machine with a radial kernel. The comparison with the decision tree can be done directly, as they use the same features, and as expected, the SVM is getting better results. As we said, the decision tree advantage is its interpretability, and we can see how being much more interpretable, the accuracy is not very far from the SVM.

The Support Vector Machine with the adaptive chi squared kernels seems to be performing worse than expected. In this case we cannot compare with the others, as the features used are different. We can observe that any of the results for the different features perform well, with a maximum accuracy of 62.70%, so, it looks probable that for our data, this kernel is not good, no matter which features we are using. Besides the bad results with this kernel, we have to consider that it is using much more features than the rest, as we have histograms as features. So, the complexity of the model is higher than in the other cases, which makes it probably the worse option of the three trained.

7.4 Parameters

The parameters used for the Decision Tree, and the Support Vector Machine with radial kernel are the number of words to consider for the features. As we can see in the results, there is no obvious correlation between the number of words used, and a better accuracy. If we consider that our dataset is relatively small, and that we performed just one validation (instead of cross validation), we could believe that with more data or more validation tests, we will find results with even less variance. So, our conclusion is that for the this problem, with the given data, the number of words used for the features is not relevant for getting good results. But we have to consider that it could be relevant with a better dataset, that should be tested.

For the adaptive chi squared kernel, we have two parameters, as explained before. Observing the results, we can again say that there is no obvious correlation between any of the parameters and the accuracy. Again, it is possible that with another dataset, and for the same problem, we could observe some correlation.

7.5 General conclusion

As a final conclusion, we can say that the lack of a good dataset, seems to be caused some poor results, and also it is probably the cause for not seeing any improvement among different methods and parameters. Our main conclusion is that a good training dataset is key for building a classifier.

References

[1] Natural Language Processing with Python, by Steven Bird, Ewan Klein and Edward Loper.

[2] Proceedings of STeP'96. Jarmo Alander, Timo Honkela and Matti Jakobsson (eds.), Publications of the Finnish Artificial Intelligence Society, pp. 64-72.