# On classifying coherent/incoherent Romanian short texts

**Anca D. Dinu**
University of Bucharest, Faculty of Foreign Languages and Literature

Strada Edgar Quinet 5-7, Bucharest, Romania
E-mail: anca_d_dinu@yahoo.com

**Abstract**

In this paper we present and discuss the results of a text coherence experiment performed on a small corpus of Romanian text from a number of alternative high school manuals.
During the last 10 years, an abundance of alternative manuals for high school was produced and distributed in Romania. Due to the large amount of material and to the relative short time in which it was produced, the question of assessing the quality of this material emerged; this process relied mostly of subjective human personal opinion, given the lack of automatic tools for Romanian.
Debates and claims of poor quality of the alternative manuals resulted in a number of examples of incomprehensible / incoherent paragraphs extracted from such manuals. Our goal was to create an automatic tool which may be used as an indication of poor quality of such texts.
We created a small corpus of representative texts from Romanian alternative manuals. We manually classified the chosen paragraphs from such manuals into two categories: comprehensible/coherent text and incomprehensible/incoherent text. We then used different machine learning techniques to automatically classify them in a supervised manner. Our approach is rather simple, but the results are encouraging.

## 1. Introduction

During the last 10 years, an abundance of alternative manuals for primary and high school was produced and distributed in Romania. Due to the large amount of material and to the relative short time in which it was produced, the question of assessing the quality of this material emerged; this process relied mostly of subjective human personal opinion, given the lack of automatic tools for Romanian.
Debates and claims of poor quality of the alternative manuals resulted in a number of examples of incomprehensible / incoherent paragraphs extracted from such manuals. Our goal was to create an automatic tool for categorization of short Romanian text, which may be used as an indication of poor quality of such texts.
The typical text categorization criteria comprise categorization by topic, by style (genre classification, authorship identification), by expressed opinion (opinion mining, sentiment classification), etc. Very few approaches consider the problem of categorizing text by degree of coherence, as in (Miller, 2004).
We created a small corpus of representative texts from 6 Romanian alternative manuals. We manually classified the chosen paragraphs from such manuals into two categories: comprehensible/coherent text and incomprehensible/incoherent text. We then used different machine learning techniques to automatically classify them in a supervised manner.
There are many qualitative approaches related to coherence that could be applied to English language. For example, segmented discourse representation theory (Lascarides, 2007) is a theory of discourse interpretation which extends dynamic semantics by introducing rhetorical relations into the logical form of discourses. A discourse is coherent just in case: a) every proposition is rhetorically connected to another piece of discourse, resulting in a single connected structure for the whole discourse; b) all anaphoric expressions/relations can be resolved. Maximize Discourse Coherence is a guiding principle. In the spirit of the requirement to maximize informativeness, discourses are normally interpreted so as to maximize coherence. Other examples of qualitative approaches related to coherence are latent semantic analysis (Dumais et al., 1988), lexical chains (Hirst & St.-Onge, 1997), centering theory (Beaver, 2004), discourse representation theory (Kamp & Reyle, 1993), veins theory (Cristea, 2003), etc.
Nevertheless, because of the lack of appropriate tools for Romanian language, we had to choose a quantitative approach for automatically categorizing short Romanian text into coherent /comprehensible and incoherent /incomprehensible. An important question for such categorization is: are there any features that can be extracted from these texts that can be successfully used to categorize them? We propose a quantitative approach that relies on the use of ratios between morphological categories from the texts as discriminant features. We supposed that these ratios are not completely random in coherent text.
Our approach is rather simple, but the results are encouraging.

## 2. The corpus

We created a small corpus of texts from 6 Romanian alternative manuals with different authors. We used 5 annotators to manually classify the chosen paragraphs from such manuals into two categories: comprehensible /coherent text (the positive examples) and incomprehensible /incoherent text (the negative examples). We selected 65 texts (paragraphs) which were unanimously labelled by all the annotators as incoherent /incomprehensible.
As some annotators observed, the yes or no decision was

overly restrictive; they could have gave a more fine grained answer such as *very difficult to follow*, *easy to follow*, etc, but we decided to work with 2 class categorisation from reasons of simplicity. We leave this for further work, as well as creating a larger corpus.
We also selected 65 coherent / comprehensible texts from the manuals, by the same method.

## 3. Categorization experiments and results

We used Balie system developed at Ottawa University http://balie.sourceforge.net/), which has a part of speech tagger for Romanian, named QTag. We only took in consideration 12 parts of speech. We eliminated the punctuation tags and we mapped different subclasses of pos into a single unifying pos (for example all subclasses of adverbs were mapped into a single class: the adverbs, all singular and plural common nouns were mapped into a single class: common nouns, etc).We manually corrected the tagging, because of the poor accuracy obtained by the parser and because the size of the corpus allowed us to do so. We computed the pos frequencies in each of the training set texts (both from the positive and from the negative examples). We normalized them (divided the frequencies to the total number of tagged words in each text), to neutralize the fact that the texts had different lengths. We then computed all possible 66 ratios between all 12 tags. In the process of computing these ratios we added a small artificial quantity (equal to 0.001) to both the numerator and the denominator, to guard against division by zero. These 66 values become the features on which we trained 3 out of 5 types of machines we employed (the other two needed no such pre-processing).
Because of the relative small number of examples in our experiment, we used leave one out cross validation (l.o.o.) (Efron & Tibshirani, 1997; Tsuda, 2001), which is considered an almost unbiased estimator of the generalization error. Leave one out technique consists of holding each example out, training on all the other examples and testing on the hold out example.
The first and the simplest technique we used was the linear regression (Duda et al., 2001; Chen et al., 2003; Schroeder et al., 1986), not for its accuracy as a classifier, but because, being a linear method, it allows us to analyze the importance of each feature and so determine some of the most prominent features for our experiment of text categorization. We also used this method as a base line for the other experiments.
For a training set:

$$S = (x_1, y_1), (x_2, y_2), ..., (x_l, y_l),$$

the linear regression method consists in finding the real linear function (i.e finding the weights w)

$$g(x) = \sum_{i=1}^{l} w_i x_i$$

such that

$$\sum_{i=1}^{l} (y_i - g(x_i))^2$$

is minimized. If the matrix X'X is invertible, then the solution is $w = (X'X)^{-1}X'y$. If not (the matrix $X'X$ is singular), then one uses the pseudo-inverse of the matrix $X'X$, thus finding the solution w with the minimum norm.
For this experiment we used the pre-processed data as described above. Its l.o.o accuracy was of 67.48%, which we used further as baseline for next experiments.
We ordered the 66 features (pos ratios) in decreasing order of their coefficients computed by performing regression. The top 5 features that contribute the most to the discrimination of the texts are linguistically very interesting:

- the ratio between the pre-determiner (such as all, this, such, etc) and adverbs, representing 15.8% of all feature weights;
- the ratio between modal auxiliary verbs and adverbs, representing 13.29% of all feature weights;
- the ratio between pre-determiner and conjunction, representing 9.10% of all feature weights;
- the ratio between modal verbs and conjunctions, representing 7.25% of all feature weights;
- the ratio between common nouns and conjunctions, representing 6.98% of all feature weights.

These top 5 features accounted for more than 50% of data variation.
The second ratio may be explained by the inherent strong correlation between verbs and adverbs. The presence of conjunction in 3 out of the top 5 ratios confirms the natural intuition that conjunction is an important element with regard to the coherence of a text. Also, the presence of the pre-determiners in the top 5 ratios may be related to the important role coreference plays in the coherence of texts.
As we said, we used the linear regression to analyze the importance of different features in the discrimination process and as baseline for state of the art machine learning techniques. Next, we tested two kernel methods (Müller et al., 2001; Schölkopf & Smola, 2002): ν support vector machine (Saunders et al., 1998) and Kernel Fisher discriminant (Mika et al., 1999; Mika et al.,2001), both with linear and polynomial kernel.
Kernel-based learning algorithms work by embedding the data into a feature space (a Hilbert space), and searching for linear relations in that space. The embedding is performed implicitly, that is by specifying the inner product between each pair of points rather than by giving their coordinates explicitly.
Given an input set $X$ (the space of examples), and an embedding vector space $F$ (feature space), let $\phi : X \rightarrow F$ be an embedding map called feature map.
A kernel is a function $k$, such that for all $x, z$ in $X$,

$$k(x, z) = < \phi(x), \phi(z) >,$$

where $< . , . >$ denotes the inner product in F.
In the case of binary classification problems, kernel-based

learning algorithms look for a discriminant function, a function that assigns *+1* to examples belonging to one class and *-1* to examples belonging to the other class. This function will be a linear function in the space *F*, so it will have the form:

$$f(x) = sign(< w, \phi(x) > + b),$$

for some weight vector *w*. The kernel can be exploited whenever the weight vector can be expressed as a linear combination of the training points, $\sum_{i=1}^{n} \alpha_i \phi(x_i)$, implying that f can be expressed as follows:

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i k(x_i, x) + b).$$

Various kernel methods differ by the way in which they find the vector *w* (or equivalently the vector α). Support Vector Machines (SVM) try to find the vector *w* that defines the hyperplane that maximally separates the images in *F* of the training examples belonging to the two classes.

Kernel Fisher Discriminant (KFD) selects the w that gives the direction on which the training examples should be projected such that to obtain a maximum separation between the means of the two classes scaled according to the variances of the two classes in that direction.

The kernel function captures the intuitive notion of similarity between objects in a specific domain and can be any function defined on the respective domain that is symmetric and positive definite. The optimization problems are solved in such a way that the coordinates of the embedded points are not needed, only their pairwise inner products which in turn are given by the kernel function *k*. Details about SVM and KFD can be found in (Taylor and Cristianini, 2004; Cristianini and Taylor, 2000).

The ν support vector classifier with linear kernel ($k(x, y) =< x, y >$) was trained, as in the case of regression, using the pre-processed 66 features, exactly the same features used for linear regression.

The parameter ν was chosen out of nine tries, from 0.1 to 0.9, the best performance for the SVC being achieved for ν = 0.4. The l.o.o. accuracy for the best performing ν parameter was 77.34%, with 9.86% higher then the baseline.

The Kernel Fisher discriminat with linear kernel was trained on pre-processed data as it was the case with the regression and ν support vector classifier. Its l.o.o. accuracy was 74.92 %, with 7.44 % higher than the baseline.

The flexibility of the kernel methods allows us to directly use the pos frequencies, without computing any pos ratios. That is, the polynomial kernel relies on the inner product of all features: it implicitly embeds the original feature vectors in a space that will contain as features all the monomial (up to the degree of the polynomial used) over the initial features. For a polynomial kernel of degree 2 for example, the implicit feature space will contain apart of pos frequencies, all the products between these frequencies, these products playing the same role as the ratios.

The support vector machine with polynomial kernel was trained directly on the data, needing no computation of ratios. The kernel function we used is:

$$k(x, y) = (< x, y > + 1)^2$$

The l.o.o. accuracy of the support vector machine with polynomial kernel for the best performing ν = 0.4 parameter was 81.13%, with 13.65% higher than the baseline.

The Kernel Fisher discriminant with polynomial kernel was trained directly on the data, needing no ratios. Its l.o.o. accuracy was 85.12%, with 17.64% higher then the baseline.

All machine learning experiments were performed in Matlab, or using Matlab as interface (Chang and Lin, 2001).

We summarized these results in the next table.

| Learning method type | Accuracy |
|---|---|
| Regression | 67.48% |
| linear Support Vector Classifier | 77.34% |
| quadratic Support Vector Machine | 81.13% |
| polynomial Kernel Fisher discriminant | 85.12% |

Table 1: Accuracy of the learning methods.

As one can see from table 1, the best performance was achieved by the Kernel Fisher discriminant with polynomial kernel, with a l.o.o. accuracy of 85.12%.

## 4. Conclusions

The best l.o.o. accuracy we obtained, i.e. 85.12% is a good accuracy because using only the frequencies of the parts of speech in the texts disregards many other important features for text coherence, such as, for example, the order of phrases, coreferences resolution, rhetorical relations, etc.

Further work: the two class classification, in the case of Romanian alternative high school manuals, is a rather dramatic classification. It would be useful to design a tool that produces as output not just a yes/no answer, but a score or a probability that the input (text) is in one of the two categories, such that a human expert may have to judge only the texts with particular high probability to be in the class of incoherent texts.

## 5. Acknowledgements

Research supported by PNII-IDEI, project 228 and University of Bucharest.

## 6. References

David, B. (2004) The Optimization of Discourse Anaphora. *Linguistics and Philosophy 27(1)*, pp. 3-56.

Cristea, D. (2003): The relationship between discourse structure and referentiality in Veins Theory, in W. Menzel and C. Vertan (eds.) *Natural Language Processing between Linguistic Inquiry and System Engineering*, „Al.I.Cuza" University Publishing House, Iaşi.

Cristianini and J. Shawe-Taylor (2000) *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Chih-Chung Chang and Chih-Jen Lin (2001) LIBSVM: a library for support vector machines. Software avalable at http://ww.csie.ntu.edu.tw/ cjlin/libsvm.

Chen, X., Ender, P., Mitchell, M. and Wells, C. (2003). *Regression with SPSS* http://www.ats.ucla.edu/stat/ spss/webbooks/reg/default.htm .

Duda, R.O., Hart, P.E., Stork, D.G. (2001) *Pattern Classification* (2nd ed.). Wiley-Interscience Publication.

Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. (1988) Using Latent Semantic Analysis to improve access to textual information. In Human Factors in Computing Systems, in *CHI'88 Conference Proceedings (Washington, D.C.)*, pages 281- 285, New York, May. ACM.

Efron and R.J. Tibshirani (1997) Improvements on cross-validation: the .632+ bootstrap method. *J. Amer. Statist. Assoc*, 92:548–560.

Hirst, Graeme and David St.-Onge (1997) Lexical chains as representation of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *Wordnet: An electronic lexical database and some of its applications*. MIT Press, Cambridge, pages 305-332.

Lascarides, A., Asher, N. (2007) Segmented Discourse Representation Theory: Dynamic Semantics with Discourse Structure, in H. Bunt and R. Muskens (eds.) *Computing Meaning: Volume 3*, pp87--124, Springer.

Kamp, H. and Reyle, U. (1993) *From Discourse to Logic. An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Kluwer Academic Publishers, Dordrecht Netherlands.

T. Miller (2004) Essay Assessment with Latent Semantic Analysis. *Journal of Educational Computing Research 28.*

S. Mika, G. Rätsch, J.Weston, B. Schölkopf, and K.-R. Müller (1999) Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE.

Mika, A.J. Smola, and B. Schölkopf (2001) An improved training algorithm for kernel Fisher discriminants. In T. Jaakkola and T. Richardson, editors, *Proceedings AISTATS 2001,* pages 98–104, San Francisco, CA.

K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf (2001) An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks, 12 (2)*:181–201.

C. Saunders, M.O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A.J. Smola (1998) Support vector machine reference manual. *Technical Report CSD-TR-98-03*, Royal Holloway University, London.

Schölkopf and A.J. Smola (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

Schroeder, Larry D., David L. Sjoquist, and Paula E. Stephan (1986) Understanding regression analysis: An introductory guide. Thousand Oaks, CA: Sage Publications. *Series: Quantitative Applications in the Social Sciences, No. 57*

John S. Taylor and Nello Cristianini (2004) *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA.

K. Tsuda, G. Rätsch, S. Mika, and K.-R. Müller (2001) Learning to predict the leave-oneout error of kernel based classifiers. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks — ICANN'01*, pages 331–338. Springer Lecture Notes in Computer Science, Vol. 2130.