ECHO at the LSHTC Pascal Challenge 2

Christophe Brouard
UPMF-Grenoble 2/CNRS
Université de Grenoble
LIG UMR 5217/AMA team
UFR IM2AG - BP 53 - F-38041 Grenoble Cedex 9, France
Christophe.Brouard@iut2.upmf-grenoble.fr

Abstract. A classification system called ECHO has been designed for our participation to the first LSHTC Pascal challenge. This system is based on a principle of echo. It computes the score of a document for a class by combining a bottom-up and top-down propagation of activation in a very simple neural network. For this second edition of the challenge, the system has been refined in order to improve its performances. The main improvement concerns the definition of a method of calibration integrating the distribution of the echo scores of the training documents for each class. Another improvement concerns the management of the relative importance of bottom-up and top-down propagation which we make vary with the class.

Keywords: classification, neural network, spreading activation method.

1 Introduction

This year, the second edition of the Large Scale Hierarchical Text Classification challenge[1] proposed three new datasets. The three datasets are larger than the previous edition datasets and the documents are this time multi-labeled (a document can belong to several classes). One dataset is extracted from the Open Directory Project[2]. The two others (one "small" and one large) are extracted from Wikipedia[3].

In the next section, we present the ECHO system designed and used last year for the first edition of the challenge [2]. In the third and fourth sections we describe the two improvements we have made this year. In the fifth section, we present the results we have obtained on the datasets of the second edition of the challenge showing the impact of our improvements. In the sixth section, practical considerations concerning the execution times are presented. In the last section, a few perspectives of this work are given.

[1] http://lshtc.iit.demokritos.gr

[2] http://www.dmoz.org/

[3] http://www.wikipedia.org/

2 ECHO Description

2.1 General Description and Related Works

We called ECHO the system we designed in order to participate to the first LSHTC challenge. This system is based on a spreading activation method and on the computation of an echo between a document and a particular class in a very simple neural network. In its general principles, this system can be compared with ART neural networks [5] although in ECHO the analogy with the neural system is much less developed. However, we keep the architecture of the network, i.e., two layers linked by oriented connections, one for representing the inputs, the other one for representing the classes. We keep also the notion of Hebbian learning (reinforcement of connections between nodes simultaneously activated). Lastly, we consider a kind of echo (or resonance), comparing a back-propagated activation with the initial pattern activation corresponding to an input and selecting the category which maximizes the similarity of these activations. It has been shown in [1] that it is possible to make an analogy between resonance and relevance in the context of Information Retrieval. Indeed, in Information Retrieval, relevance is often formalized by two implications between the document and the query. The first implication is oriented from the document to the query and a second one is oriented from query to the document. It corresponds to the well known specificity and exhaustivity aspects found in most of the models of relevance. The analogy proposed in [1] consists in representing both implications by a bottom-up and a top-down propagation of activation. Thus, this system is based on a model of relevance and can be seen as a model of information selection. In the context of a classification task, the selection consists in choosing the good class. This system could also be applied to other problem of information selection. The idea of information selection based on spreading activation methods in networks is not new. In the context of semantic networks, numerous methods have been developed and even if the idea seems interesting, the results have not been always very satisfying [3]. Numerous problems have been encountered. One of these concerns the artificial constraints necessary to avoid the activation of all the nodes of the network. We believe that the concept of echo can be a natural way of controlling spreading activation.

2.2 Network Construction (Training)

In the training phase, the system connects the nodes representing the terms of the documents to the nodes representing the classes of these documents. The connections between the nodes are oriented and weighted (Fig. 1). The weight w_{ij} of the connection from a node j to a node i is based on the relative frequency f_{ij} of the concept represented by i given the concept represented by j. For the calculation of these frequencies, considering that a term can be more or less present in a document depending on the frequency of the term, we consider degrees of membership [6]. Let

$\mu_D(i)$ denotes the degree of membership in the real unit interval [0,1] of a term or a class i to a document D. Then, the frequency is defined as follows:

$$f_{ij} = \frac{\sum_D \mu_D(i).\mu_D(j)}{\sum_D \mu_D(j)} \tag{1}$$

More precisely, the weight corresponds to this relative frequency divided by the sum of the relative frequencies of all the concepts i given the concept represented by j as follows:

$$w_{ij} = \frac{f_{ij}}{\sum_i f_{ij}} \tag{2}$$

Thus, the sum of the output connections weights of every node is equal to 1. For example, if i is a class and j is a term, the relative frequency f_{ij} of i given j corresponds to the number of documents of the class i containing the term j divided by the number of documents containing the term j. The weight w_{ij} corresponds to this relative frequency divided by the sum of the relative frequencies of all the classes given the term j. This sum is different from 1 when the documents can belong to several classes.

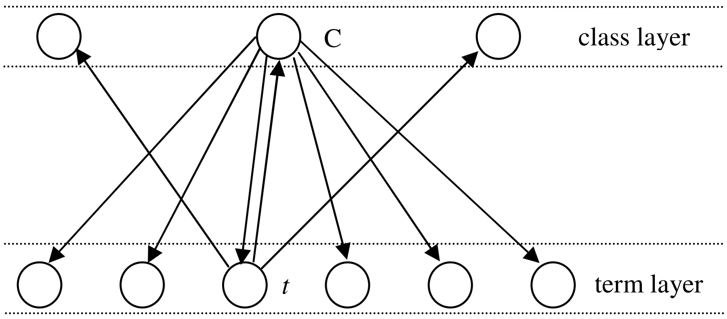

Fig. 1. The built network. The connections are oriented from a term to a class or from a class to a term. For readability reasons, only the connections of a particular term t and a particular class C are represented in the figure.

As we mentioned above, a membership function must be defined. Concerning the class nodes, the situation is binary and the solution is simple, i.e., the membership degree is 1 when the document is in the class, the degree is 0 when it is not. Concerning the term nodes, after different tests we have chosen to define it as follows:

$$\mu_D(t) = \min\left(1, pm.\frac{nb\_occ(D,t)}{nb\_terms(D)}\right) \tag{3}$$

where *nb_occ(D,t)* corresponds to the number of occurrences of the term *t* in the document D, *nb_terms(D)* corresponds to the number of terms in the document D and *pm* is a positive integer (a parameter of the system). So if the proportion of the term *t* is larger than 1/*pm*, then the membership degree is 1 (we consider that the term is totally present). Otherwise, the membership degree corresponds to the ratio between the term frequency with 1/*pm*.

2.3 Network Utilization (Classification)

In the classification phase, each new document is represented by an activation pattern in the term layer, i.e., the nodes representing the terms of the document are activated. The activation spreads in the built network and a degree of echo between the considered document D and each class C is computed. This degree of echo measures the quantity of activation returned to the initial activation pattern representing the document D after propagation to C and a back-propagation from C to the term layer. This degree of echo is defined as follows:

$$\mathrm{E(D, C)} = \sum_{\mathrm{t \in D}} \mathrm{a(t)}.\left[\mathrm{W_{tC}} . \sum_{\mathrm{t \in D}} \mathrm{a(t).W_{Ct}} \right] \quad (\mathbf{4})$$

where *a(t)* denotes the initial activation of the term *t* and depends on the frequency of the term in the document. This formula simply corresponds to the dot product between the vector representing the document (initial activation pattern) and the vector representing the activation pattern in the term layer after propagation to C and back-propagation from C. In this calculation we always consider that the activation of a node is spread and divided according to the weight of the connections. We also consider that the activation of a node corresponds to the sum of the activations received from the connected nodes.

As we mentioned above, an activation function must be defined. After different tests we have chosen to use the same kind of function as for the membership degree. So, the activation of a term *t* for a document D is defined as follows:

$$a(t) = \min\left(1, pa . \frac{nb\_occ(D,t)}{nb\_terms(D)} \right) \quad (\mathbf{5})$$

where *nb_occ(D,t)* corresponds to the number of occurrences of the term *t* in the document D, *nb_terms(D)* corresponds to the number of terms in the document D and *pa* is a positive integer (a parameter of the system). So if the term frequency of the term *t* is larger than 1/*pa*, then the activation is 1. Otherwise, the activation corresponds to the ratio between the document frequency of the term with 1/*pa*.

3 A Method of Calibration Integrating the Distribution of the Scores

3.1 Idea

The most obvious way to take the scores *E(D,C)* into account in order to decide which class to assign to a document consists in choosing the class corresponding to the largest score. This is the strategy we adopted for the first challenge.
However, it appears that the distribution of the scores depends on the class (Fig. 2) and that a same score can be interpreted differently for two different classes depending on the fact that training documents with a similar score belong or do not belong to the class.

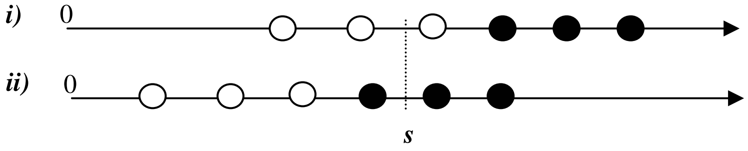

Fig. 2. The cases *i)* and *ii)* correspond respectively to the score distributions of two classes. The black circles correspond to the score of the documents which belong to the class. The white circles correspond to the score of the documents which do not belong to the class. As we can see, even if a document has the same score *s* for both classes, the score distributions are in favour of the class corresponding to the case *ii)*.

Thus, we propose to measure the quality of the position of the score *s* in the distribution of the scores for the class *C* trying to estimate a probability to belong to the class given a score. This problem can be seen as a problem of calibration. See [4] for a review of existing methods of calibration.

3.2 Formalization

Various ways of taking into account the distribution of the scores have been tested. The best results have been obtained by combining three quantities:

- The number of documents with a larger score than *s* which do not belong to the class *C* denoted by nb_{sC}.
- The number of documents with a smaller score than *s* which belong to the class *C* denoted by bs_{sC}.
- The number of documents with a larger score than *s* which belong to the class *C* denoted by bl_{sC}.

These three quantities are combined in a function *F(s,C)* giving for each score *s* and class *C*, a value in [0,1] that expresses the quality of the position of the score *s* in the distribution of the scores for the class *C*. *F(s,C)* is defined as follows:

$$F(s,C)=\begin{cases} 1 & \text{if } nb_{sC}=0 \\ \dfrac{(bs_{sC}+cl.bl_{sC})}{cn.nb_{sC}+(bs_{sC}+cl.bl_{sC})} & \text{otherwise} \end{cases} \quad \textbf{(6)}$$

where *cn* is a coefficient allowing to vary the relative importance of nb_{sC} versus bs_{sC} and bl_{sC} and where *cl* is a coefficient allowing to vary the relative importance of bl_{sC} versus bs_{sC}. This measure can be seen as an estimation of the probability to belong to the class *C* given a score *s*.

3.3 Combination of E(D,C) and F(s,C)

Various ways of combining *E(D,C)* and *F(s,C)* have been tested. The best results have been obtained by computing a new score *S(D,C)* as follows:

$$S(D,C)=E(D,C).F(E(D,C),C) \quad \textbf{(7)}$$

It corresponds to a simple product between the echo score *E(D,C)* and the value of *F* for this score. Finally, we decided to assign the document to the class with the best *S(D,C)* score.

$$Decision(D)=\underset{C}{ArgMax}\,(S(D,C)) \quad \textbf{(8)}$$

3.4 Implementation

The new way to compute the score requires two steps of calculation. In a first step the echo scores *E(D,C)* of all the training documents for the different classes are computed. More precisely, for each training document, after removing the document from the training set, the *N* best scores plus the scores of the classes to which the document belongs to (if they are not included in the *N* best) are registered in the score distributions of the different classes. In our experiments, we chose *N*=25. In a second step, the "echo scores" *E(D,C)* of the documents of the test set are computed and the distributions built in the first step are used in order to compute *F(D,C)* and then *S(D,C).*

4 Adapting the Relative Importance of Top-down and Bottom-up Propagation to the Class

4.1 Idea & Formalization

The formula 4 can be rewritten as follows:

$$\mathrm{E(D,C)} = \left[\sum_{t\in D} a(t).W_{tC}\right].\left[\sum_{t\in D} a(t).W_{Ct}\right] \qquad \mathbf{(9)}$$

This new formulation makes appear two different factors. The first factor corresponds to the top-down pathway from C to D. It measures the presence of the terms which are the most frequent in the documents of this class. In other words, this first factor checks for the presence of expected terms. The second factor corresponds to the bottom-up pathway from D to C. Its nature is very different. It focuses on the presence of terms specific to the class C. This second factor can be seen as a rule-based system in which each connection corresponds to a rule and each term votes for classes with a strength depending of term activation and its output connections weights. The two factors correspond also to the exhaustivity and specificity aspects which can be found in most of the relevance models in Information Retrieval [1]. We give the possibility to vary the importance of each factor by introducing a new parameter *p* as follows:

$$\mathrm{E(D,C)} = \left[\sum_{t\in D} a(t).W_{tC}\right]^{p}.\left[\sum_{t\in D} a(t).W_{Ct}\right]^{2-p} \qquad \mathbf{(10)}$$

Previous experiments have shown that when classes are not homogeneous, i.e., documents which belong to a same class are not similar, the relative importance of the first factor (and consequently the *p* parameter) must be decreased. This can be explained by the fact that in this case the term expectation is less important. Without analyzing the reasons of *p* variation, one can ask if the optimal value of *p* varies with the considered subset of classes. Then, a new problem arises: how can we compare *E(D,C)* scores computed with different values of *p*? In a first step we tried to vary the value of *p* ignoring the problem of comparison.

4.2 Implementation

The set of the all the classes have been divided into little subsets of contiguous classes according to the hierarchy description. The optimal value of *p* is computed on the training set for each subset. For the test, the score of the documents are computed for each class with the value of *p* corresponding to the subset to which the class belongs.

5 Results and Discussion

We submitted results for DMOZ (Open Directory) and WIKIPEDIA SMALL datasets. In the case of DMOZ, we kept the parameters values found for the dry-run dataset of the previous edition of the challenge (*pa*=20, *pm*=60, *cn*=2, *cl*=0.2) . In the case of wikipedia, we tried different values on the training set and we obtained the best results for *pa*=30, *pm*=30, *cn*=2, *cl*=0.2. In both cases, the first improvement (calibration) has a strong effect corresponding to about 4% of accuracy. Concerning the second improvement (variation of *p*), an important difference between optimal values of *p* for the different subsets of classes have been observed for the DMOZ training dataset (see table 1). This difference has been exploited to improve the results. At the contrary, no difference has been observed for the WIKIPEDIA SMALL dataset for which the optimal value of *p* is 1.4.

Table 1. Optimal decomposition of the space of the classes and corresponding *p* value for DMOZ dataset.

class number interval	**[0-6000[**	**[60 000-120 000[**	**>=120 000**
optimal value of *p*	1.6	1.0	1.6

Concerning the combination of improvements (which can only be tested in the case of the DMOZ dataset), the best result is obtained when both improvements are combined. However, the improvements are not simply added (see table2).

Table 2. Accuracy of the system with or without improvements for the DMOZ dataset. *Base* corresponds to the system without improvement. *I1* corresponds to the first improvement (score distributions integration) and *I2* corresponds to the second improvement (variation of *p*).

	Base	**Base+I1**	**Base+I2**	**Base+I1+I2**
Accuracy	0.3407	0.3840	0.3553	0.3885

Indeed, the first improvement corresponds to a gain of 0.04, the second improvement corresponds to a gain of 0.015 whereas the combination of both improvements corresponds only to a gain of 0.0485. In order to explain the difference between the result we expected (addition of improvements) and the result we observed, different experiments have been conducted. These experiments showed that a strong dependency between *p* parameter of the first improvement and *cn, cl* parameters of the second improvement is responsible for the difficulty to combine efficiently both improvements. Thus, the *cn* and *cl* parameters are not optimal for all the values of *p*. At last, the variation of the *cn* and *cl* with *p* is not a solution since our experiments showed that in this case the scores cannot be compared anymore.

6 Practical Considerations

The system is implemented in Java. It is run on a PC with an Intel Core i5 2.53Ghz CPU with 2Go of RAM. The limitation of the RAM leads to divide the space of the classes into 3 subsets and to fusion the results after. This decomposition leads also to an increase of computing times which are reported in table 3. We can see that the cost of the score distributions construction is the most important. It corresponds roughly to the classification time of the training documents set which is respectively 3.75 and 5.6 times larger than the test documents set.

Table 3. Times in seconds of the training and classification phases for the DMOZ and WIKIPEDIA SMALL datasets.

network construction		**distributions construction**		**classification**	
dmoz	wikipedia	dmoz	wikipedia	dmoz	wikipedia
1147	361	39785	17485	9895	2897

7 Conclusions

The comparison of our results with the results obtained by the other participants (in particular on the DMOZ dataset) are pretty good. This shows that a system based on a simple spreading activation method can result in a high effectiveness. The variation of the *p* parameter is a central aspect of the system. This parameter should be more precisely studied and correlated with general characteristics of the datasets (like homogeneity of classes). Moreover, an automatic method for the segmentation of the space of the classes could also be defined and solutions for the comparison of scores and for the combination of calibration and variation of *p* could also be found.

References

1. Brouard, C., Nie, J.Y.: Relevance as Resonance: a New Theoretical Perspective and a Practical Utilization in Information Filtering. Inf. Proc. & Manag. 40, 1--19 (2004)
2. Brouard, C.: Classification by resonance in an associative network. Large Scale Hierarchical Text Classification Workshop of ECIR (2010)
3. Crestani, F.: Application of Spreading activation Techniques in Information Retrieval, Artificial Intelligence Review, 11, 453--482 (1997)

4 Gebel, M.: Multivariate calibration of classifier scores into the probability space, Dissertation, Universität Dortmund (2009)

5. Grossberg, S.: Adaptive pattern classification and universal recoding. I. Parallel development and coding of neural feature detectors. Biol. Cyber., 23, 117--140 (1976)
6. Zadeh, L.A.: Fuzzy Sets. Information and Control, 8, 338--353 (1965)