Text Classification: A Sequential Reading Approach

Gabriel Dulac-Arnold, Ludovic Denoyer, Patrick Gallinari

LIP6 - University Pierre et Marie Curie - Paris

April 20, 2011
Outline

- Motivation
- Textual Classification as a Sequential reading process
- Learning
- Experiments
- Conclusion and Perspectives
Machine Learning for Text Classification has not changed much since early 2000.

- Entire documents are passed into a classifier with a bag-of-words representation.
- Many *Bag-of-Words* approaches (SVM, Naive Bayes)

Inconveniences:

- Lacks ability to consider only subsections of the document at hand.
- Has no ability to gauge if available information is sufficient.
- Cannot act with real-time interaction, or streams of text.

Very different from a natural approach.
Contributions

Idea:
- Define a classification agent.
- This agent reads a document sequentially, and decides when to classify.

Our contribution in three steps:
- Model TC as a Sequential Decision Process.
- Model our Sequential Decision Process as an MDP and use Reinforcement Learning to find a good policy.
- Apply this approach to a couple of TC corpora and get some interesting results.
Problem Definition

- Training Documents $\mathcal{D} = (d_1, \ldots, d_N)$.
- Classes $\mathcal{C} = (c_1, \ldots, c_L)$.
- Multi-label and Mono-label classification.
- Learning Criterium: Maximise the Average $F_1$ Score.
  - Equivalent to accuracy in the mono-label case.

Classical approaches consider the document to be a TF-IDF vector passed to a standard vectorial classifier.
TC: A Sequential Decision Problem

Let us consider TC as a Sequential Decision Problem:

- A textual document is a series of sentences: $d = (\delta^d_1, \ldots, \delta^d_n)$.
- Reading the sentences one-by-one is a sequential process.
  - Structured textual data (XML) can also be parsed in a sequential manner.
- Classification can happen at any point in the reading process.

Modelling TC as a Sequential Decision Process allows the system to classify without reading the whole document.
Example of Inference in the SDP

The dry period means the temporoa will be late this year. Again it seems that cocoa delivered earlier on consignment was included in the arrivals figures. In view of the lower quality over recent weeks farmers have sold a good part of their cocoa held on consignment. Comissaria Smith said spot bean prices rose to 340 to 350 cruzados per arroba of 15 kilos.
Markov Decision Process

We model our system as a deterministic Markov Decision Process: $(S, A, T, r)$

- State $S$, Actions $A$
- Transition Function $T : S \times A \rightarrow S$
- Reward Function: $r : S \times A \rightarrow \mathbb{R}$
- Policy $\pi : S \rightarrow A$

Total Reward on an Episode:

$$R^\pi(s) = \mathbb{E}_{P_\pi} \left[ \sum_{t=0}^{N} r(s_t, \pi(s_t)) | s_0 = s, a \right].$$

Optimal Policy: $\pi^* = \arg\max_{\pi} \mathbb{E}_{s \sim D}[R^\pi(s)]$
Formalisation as an MDP

- Each state is a tuple \((p, \hat{y})\):
  - \(p \in [1, n_d]\) is the current sentence being read; this implies that \(\delta^d_1\) to \(\delta^d_{p-1}\) have already been read.
  - \(\hat{y}\) is the set of previously assigned categories where \(\hat{y}_k = 1\) iff the document has been assigned to category \(k\), 0 otherwise.

- Actions : \(A = C_a \cup \{N\}\)
  - **Classification** actions denoted *classify as* \(k\).
  - **Next sentence** action denoted *next*.
  - **Stop** action denoted *stop*.

- Reward at state \(s\):
  \[
r(s, a) = \begin{cases} 
  F_1(\hat{y}, y) & \text{if } a \text{ is a *stop* action} \\
  0 & \text{otherwise}
  \end{cases}
  \]
Illustration of the MDP

- Classify as *acq*
- Classify as *cocoa*
- Simulating the current policy
- Computing the classification loss at the end of the process
- Labeling the actions as good action (+1) and bad action (-1)

- The actions are used as training examples for a classical classifier

- The classifier learns from the training examples

- The classifier makes predictions on new data

- The predictions are used to guide the decision-making process

- The process repeats until the desired outcome is achieved

- The actions are used as training examples for a classical classifier
Learning Algorithm

We used a Monte Carlo approach to generate training examples.

Policy Iteration Algorithm

- Generate training states.
- Use $\pi_{t-1}$ to estimate expected return of all actions in each state and generate training set for classifier.
- Train classifier to discriminate between 'good' and 'bad' actions to obtain $\pi_t$. 
Motivation

Prerequisites

Sequential Reading Process

Results

Conclusion

Illustration of Learning

The day production was
the temporal will be
liable this year. Against
it
seems that cocoa
delivered earlier on
consignment was
included in the annual
figures. In view of the
lower quality of recent
years, farmers have sold
a good part of their
cocoa.

Consignment: Commodities
Smith said spot bean
prices rose to 340 to 350
U.S. dollars a metric ton
of 15 kilos.

Classify as
acq

Classify as
cocoa

next

stop

Loss = 0

Good action
(+1)

Loss = 0.5

Bad action
(-1)

Loss = 1

Bad action
(-1)

Loss = 1

Bad action
(-1)

Sampling a state
over the training
document

Enumerating
all the
possible
actions

Simulating the current policy

Computing the classification
loss at the end of the process

Labeling the actions as
good action (+1) and bad action (-1)

The actions are used as
training examples for a
classical classifier

Gabriel Dulac-Arnold, Ludovic Denoyer

ECIR2011
Practical Implementation

- Vector representation of the state is necessary: $\phi(s_t)$.
- The state is composed of previously read sentences, current sentence, and current labels:

$$
\Phi(s) = \begin{pmatrix}
\sum_{i=1}^{p} \delta_i^d p \\
\delta_p^d \\
\hat{y}_0 \ldots \hat{y}_c
\end{pmatrix}
$$

- Block vector representation: $\Phi(s, a) = (0 \ldots \phi(s) \ldots 0)$
Datasets

- The Reuters-21578 dataset which provides two corpora:
  - The **Reuters8** corpus is a mono-label corpus composed of 8 different categories.
  - The **Reuters10** corpus which has been built by keeping the 10 largest categories.
- The **WebKB** dataset is a corpus a mono-label corpus composed of Web pages dispatched in 4 different categories.
- The 20 **Newsgroups** dataset is a mono-label corpus of news composed of 20 classes.

<table>
<thead>
<tr>
<th>Corpus</th>
<th># docs</th>
<th># cats</th>
<th># sentences / doc.</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>R8</td>
<td>7678</td>
<td>8</td>
<td>8.19</td>
<td>Mono-label</td>
</tr>
<tr>
<td>R10</td>
<td>12 902</td>
<td>10</td>
<td>9.13</td>
<td>Multi-label</td>
</tr>
<tr>
<td>Newsgroup</td>
<td>18 846</td>
<td>20</td>
<td>22.34</td>
<td>Mono-label</td>
</tr>
<tr>
<td>WebKB</td>
<td>4 177</td>
<td>4</td>
<td>42.36</td>
<td>Mono-label</td>
</tr>
</tbody>
</table>

**Table:** Corpora statistics.
Results

Baseline performance obtained with a linear SVM using TF-IDF-weighted vector representations of each document. Metrics:

- **Macro-\(F_1\):** Unweighted average of each class’s \(F_1\) score. Small classes that are hard to classify can therefore disproportionately affect the overall score.
- **Micro-\(F_1\):** Each class’s average \(F_1\) score is weighted by the class’s relative size. This metric is therefore more representative of overall performance.
Figure: Performances over R8 (left) and NewsGroup (right)
Figure: Performances over WebKB (left) and R10 (multilabel) (right)
**Figure**: Reading size for each corpus (left) and number of documents vs reading size (right)
In summary:

- We proposed a formalisation of text classification as a sequential reading process.
- We used a Reinforcement Learning method for learning the best policy which classifies correctly while reading only part of the document.
- Experimental results show that the performances are almost equivalent to baseline bag-of-words approaches while taking into consideration much less information.
- The proposed algorithm exhibits a interesting behaviour, learning to read more when the task is more difficult.
Perspectives

A few directions being considered.

- Including a cost on the next action.
- Extending the set of possible actions to acquire knowledge from the Web...
Preliminary results

Use of an external dictionary for unknown words: 'give me the definition of...' actions.

With Dictionary - 3% train - 5% words - R8
Questions?