A Self-Pruning Classification Model for News

L. Akritidis[1], A. Fevgas[2], P. Bozanis[3], M. Alamaniotis[4]

[1,2,3]DaSE Lab – Dpt. of Electrical & Computer Engineering – Univ. of Thessaly

[4]Dpt. of Electrical and Computer Engineering – Univ. of Texas at San Antonio

10th International Conference on Information, Intelligence, Systems and Applications (IISA 2019)

July 15-17, 2018, Patras, Greece

News Aggregators

- Collect news feeds from various sources (news sites, portals, etc.).
- They offer two main services:
  - They group similar articles into the same cluster.
  - They classify the articles into a predefined set of categories/classes (politics, economics, athletics, etc.).
- We address the second problem, i.e., news articles classification.

News Articles Categorization

- One of the most important problems in the area.
- Given a news article and a set of categories, it is required that we determine the category where the article belongs to.
- It leads to numerous novel applications:
  - Query expansion and rewriting.
  - Relevant/Similar articles retrieval.
  - Personalized recommendations etc.

Theoretical Background

- Let Y be the set of all categories.
- An article x can only be assigned one category y.
- Each article x is described by its title τ.
- The title consists of a set of words $(w_1, w_2, \ldots, w_{lx})$.

N-grams vs. words

- The proposed method performs morphological analysis of the titles and extracts n-grams of variable sizes.
- The reason of employing n-grams is that a n-gram is less ambiguous than single words.
- For instance, the word *company* may be correlated with multiple diverse categories (*Economics*, *Politics*, or even *Society*).
- However, the bigram *company shares* is much more specific.

Tokens and Ambiguity

- We collectively refer to all n-grams and words as *tokens*.
- Each token has its own level of *ambiguity*.
  - Ambiguous tokens are not tightly correlated with a single category; they can be connected with multiple categories.
- Or, each token is of different importance.
- According to the previous example, longer tokens are less ambiguous, that is, more important.

Importance Scores

- Furthermore, a token is more important if it has been correlated with only a few categories.
- Based on these notifications, we introduce the following importance score for a token t.

$$I_t = l_t \log \frac{|Y|}{f_t}$$

  - $|Y|$: the total number of categories,
  - f_t: the frequency of t, i.e. the number of the categories which have been correlated with t, and
  - l_t: the length (in words) of t.

Categorization Model: Training Phase

- The training phase builds a lexicon L for the tokens. Each entry in the lexicon stores:
- The token t,
- Its frequency f_t,
- Its importance score I_t,
- A list Λ_t which for each token-category relationship, includes a pair in the form $(y, f_{t,y})$, where y is a category and $f_{t,y}$ is another frequency value that reflects how many times t has been correlated with the category y.

Model Training Algorithm

Algorithm 1: Model training algorithm

```
initialize the lexicon L;
for each article x with class y do
    extract the title τ;
    for each n ∈ [1, N] do
        compute all tokens T_n of length n from τ;
        for each token t ∈ T_n do
            if t ∈ L then
                if y ∉ Λ_t then
                    f_{t,y} ← 1;
                    Λ_t.insert(y, f_{t,y});
                    f_t ← f_t + 1;
                else
                    f_{t,y} ← f_{t,y} + 1;
                end
            else
                f_{t,y} ← 1;
                Λ_t.insert(y, f_{t,y});
                f_t ← 1;
                L.insert(t);
            end
        end
    end
end
for each token t ∈ L do
    I_t ← l_t log(|Y|/f_t);
    sort Λ_t in decreasing f_{t,y} order;
end
```

Categorization Model: Testing Phase

- The testing phase is based on the lexicon L of the previous phase.
- Initially, an empty candidates list Y' is created.
- In the sequel, for each token t of each news article x, we perform a search in L.
- In case it is successful, we retrieve Λ_t, f_t and I_t of t. Then, we traverse Λ_t and for each entry y we update the candidates list Y'.

Candidates Scoring

- The score $S_{t,y}$ of each candidate category is expressed as linear combination of the importance score of the token and a quantity $Q_{t,y}$:

$$S_{t,y} = k_1 I_t + k_2 Q_{t,y}$$

$$Q_{t,y} = \log f_{t,y} \log \frac{|Y|}{f_t}$$

- Finally, the candidates are sorted in decreasing $S_{t,y}$ order and the top candidate is selected as a category for the article.

Categorization Algorithm

Algorithm 2: Categorization algorithm

```
1  for each article x in the test set do
2      initialize candidates list Y';
3      extract the title τ;
4      perform linguistic processing of τ;
5      for each n ∈ [1, N] do
6          compute all tokens T_n of length n;
7          for each token t ∈ T_n do
8              if L.search(t) == true then
9                  for each pair (y, f_{t,y}) ∈ Λ_t do
10                     if Y'.search(y) == false then
11                         Y'.insert(y);
12                         set S_y ← f(I_{t,y}, Q_{t,y})
13                     else
14                         set S_y ← S_y + I_{t,y}
15                     end
16                 end
17             end
18         end
19     end
20     sort Y' in decreasing S_y order;
21     set y ← Y'[0];
22 end
```

Model Pruning

- We may decrease the size of the lexicon L by preserving only the tokens whose importance score exceeds a threshold C:

$$C = T \max_{t,y} I_{t,y}$$

- We will demonstrate experimentally that this choice combines a significant reduction of the size of L, with infinitesimal losses in the accuracy of the algorithm.

Experimental Setup

- Dataset: UCI News Aggregator
- 400,000 new stories & 4 categories.
- Training/Test Set sizes: 70%/30%.
- Two scenarios for the extracted n-grams:
  - N=1: Extract unigrams (single words) only.
  - N=3: Extract unigrams, bigrams, & trigrams.
- $0.0 < T < 1.0$ with steps of 0.1.

Examined Methods

- SNC – Supervised News Classifier – the proposed algorithm.
- LogReg – Logistic Regression.

Accuracy Evaluation

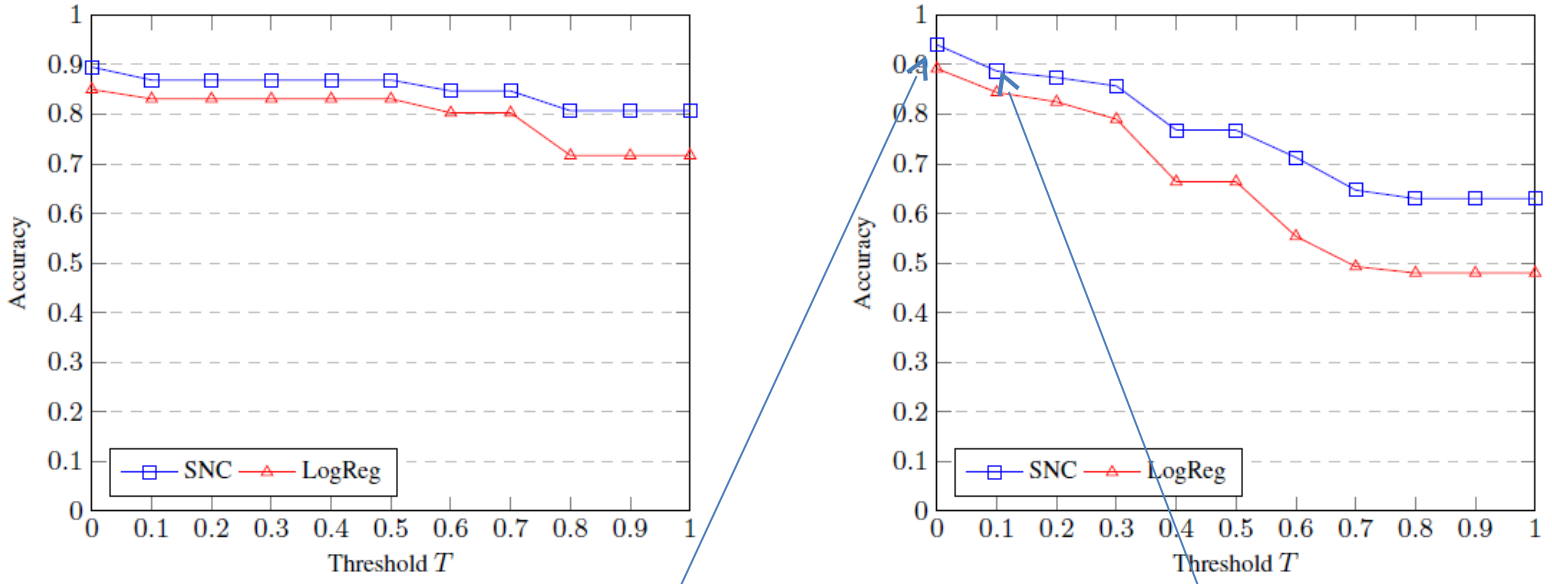

Fig. 1. Comparison of the classification performance of SNC against Logistic Regression (LogReg) for i) $N = 1$ (left), and ii) $N = 3$ (right).

Highest Accuracy: 94.1% for N=3, T=0.

Interesting Measurement: Accuracy 90% for N=3, T=0.1.

Pruned Model Evaluation

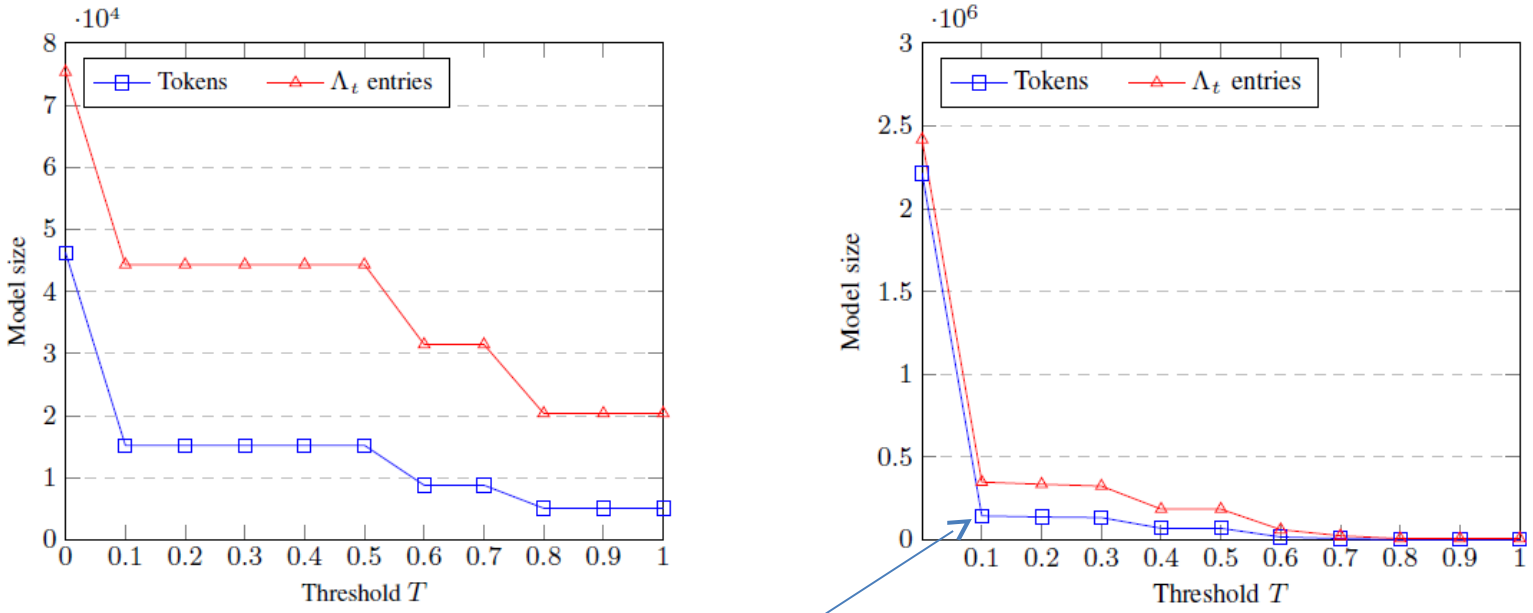

Fig. 2. Model size (overall number of tokens and Λ_t entries) reduction with respect to the value of the cut-off threshold T for: i) $N = 1$ (left), and ii) $N = 3$ (right).

Recall: For N=3, T=0.1, accuracy 90% with 16x fewer tokens and 6x fewer Λ_t entries

Conclusions

- We presented a supervised learning algorithm for news articles categorization.
- It trains a classification model based on:
  - The morphological analysis of the titles,
  - The extraction of n-grams of variable sizes,
  - The assignment of importance scores to each token.
- The method achieves ~95% classification accuracy.
- It also embodies a self-pruning strategy.

Thank You

Any Questions?