COMP90051 Project 1: Author Attribution

Group 53

April 23, 2022

1 Introduction

The problem given is an author attribution task, which requires us to identify whether an author participated in the writing of a document. We are provided with 26108 papers to train on, each one recording the venue it was published at, the year it was published, a list of specific keywords in the title and abstract, and a list of the authors of the paper. The task is to determine whether a given target author is a true co-author of a paper when provided with the true co-authors, venue, year, and keywords.

Integers are used as IDs for all values in the data instances. As mentioned in the project specification, the list of keywords only records the presence of certain keywords, as such we assume there are no duplicates and no order within the list and thus treat is as a set (the same is assumed for the list of authors). Where $[n]_0 = \{0, 1, 2, ..., n\}$, we will use the following:

- the set of all authors, $A = [2301]_0$
- the set of all venues, $V = [469]_0 \cup \{\text{“ ”}\}$
- the set of all keywords, $K = [499]_0$
- a training instance with ID n provided in train.json is (v, y, A_n, K_n) where $v \in V, A_n \subseteq A, K_n \subseteq K$ and y is a year
- a test instance with ID n provided in test.json is (v, y, A_n, K_n, t) where $v \in V, A_n \subset A, K_n \subseteq K, t \in A \setminus A_n$ and y is a year

We treated the authors as our labels in training, and since each training instance can have multiple authors, for the initial preprocessing we replaced each training instance (v, y, A_n, K_n) in the following way: for each $t \in A_n$ we create a new training instance $(v, y, A_n \setminus \{t\}, K_n, t)$, which is the same format as the test data.

2 Final approach

3 Alternatives considered

3.1 Binary representation

The training instances, even after our initial preprocessing, cannot be used as is due to them containing sets for the authors and keywords which are incompatible with most machine learning techniques. A simple solution would be to replace each possible set with an integer to represent it, however this would make it rather difficult for the model to notice trends between papers with similar but not identical coauthors without a carefully defined mapping function. This would also clearly cause an overflow since there are $2^{|A|}$ possible sets for A_n. The first solution we opted for was to represent A_n and K_n as a sequence of 0s and 1s in the following manner: given a set of coauthors A_n, we represent it as $a_n = (a_{n,0}, a_{n,1}, a_{n,2}, ..., a_{n,2301})$ where $a_{n,i} = 1$ iff $i \in A_n$, otherwise $a_{n,i} = 0$ (the same method was applied to the set of keywords too). We then joined the binary representations of the coauthors and keywords together with the publication year.

We decided to ignored the venue at which the paper was published since all papers without specific venues would be treated the same which we believed to be undesirable. As such our final mapping was:

$$\mathcal{F}((v, y, A_n, K_n, t)) = (y, a_{n,0}, a_{n,1}, a_{n,2}, ..., a_{n,2301}, k_{n,0}, k_{n,1}, k_{n,2}, ..., k_{n,499})$$

The main reason this representation was abandoned was due to the size of each training instance after the mapping being too large for many models to handle. Each instance now had $1 + |A| + |K| = 2803$ features, and with almost 50000 training instances after the initial preprocessing, the amount of time it took for models to train was too long for us to effectively use.

3.2 Padded representation

Inspection of the training data found that, while there may be over 2000 authors and 500 keywords in the entire dataset, neither the number of authors nor the number of keywords for each paper were anywhere near that amount. Thus we defined a new mapping as follows:

$$\mathcal{F}((v, y, A_n, K_n, t)) = (y, a_{n,0}, a_{n,1}, ..., a_{n,m_a}, k_{n,0}, k_{n,1}, ..., k_{n,m_k})$$

where m_a and m_k are the max number of authors and keywords in any given training instance respectively. The values of $a_{n,i}$ and $k_{n,i}$ were then simply taken as the values given in A_n and K_n respectively. This greatly reduced the number of features down to $1 + m_a + m_k$ which we found to be $1 + 9 + 109 = 119$ for the provided training set. Any instances that did not have the max number of authors/keywords were padded with the target author t and the first keyword in K_n. This representation was never tested by itself as we believed it too simplistic and naive. We attempted to modify it a little further as described below.

3.2.1 Coauthor and keyword matrices

Instead of simply taking the padded sets of coauthors and keywords, we tried to order them such that coauthors and keywords more strongly associated with the target author would be further to the left, hoping that our model would be able to recognise these features as more significant.

3.3 Research group representation