Tagging Raw Job Descriptions

Introduction

Indeed.com provides the world's largest job search engine. As engineers at Indeed, we want to give our users the best possible experience. Our current goal is to enhance the job searching process by automatically extracting interesting properties (tags) from job descriptions. These tags allow users to filter and find jobs more easily. For example, given the full job description, our program should accurately extract information like the number of required years of experience.

This challenge asks for a machine learning solution to accurately assign tags given the information in the job descriptions. We are interested in only these *twelve* tags:

- part-time-job
- full-time-job
- hourly-wage
- salary
- associate-needed
- bs-degree-needed
- ms-or-phd-needed
- licence-needed
- 1-year-experience-needed
- 2-4-years-experience-needed
- 5-plus-years-experience-needed
- supervising-job

Note that some tags are *mutually exclusive*, meaning, for example, that a job can not require both 2-4 years of experience and 5 years of experience. It is also possible that the job description does not contain any information relevant for tagging.

Dataset

We provide the zip file (*MD5* checksum is `cdd5e131a60413d0e0a6dd3da97d6cee`): indeed_ml_dataset.zip containing the following two `.tsv` (tab separated) files when unzipped:

- `train.tsv` is the training dataset providing *tags* and 4375 job descriptions. The header row has the following two columns:
  - `tags`: A space-separated list of tags.
  - `description`: A job description.
- `test.tsv` is the testing dataset providing the raw job descriptions of 2921 jobs. The header row has only column: `description` containing job descriptions.

Submission Details

You are required to upload the following three files:

- The output file, `tags.tsv` (max allowed size is `10MB`). The file should contain a space-separated list of tags -- the order of the tags does not matter -- for each of the job descriptions from the file `test.tsv` in that same order. You can choose not to tag any description by providing an empty list of tags.

A valid output file has the following format (The fourth row indicates empty tags list):

```
tags
2-4-years-experience-needed licence-needed
1-year-experience-needed associate-needed

2-4-years-experience-needed licence-needed
1-year-experience-needed licence-needed
.
.
.
part-time-job
bs-degree-needed
ms-or-phd-needed
licence-needed
```

Note that:

- The first line of the output file should contain the header, with the only column: `tags`.
- There should be exactly 2922 rows including the column header. The raw job descriptions should be tagged in the same order as given in the `test.tsv`.
- You should not use any tag other than the *twelve* tags provided.

- A *PDF* file (maximum allowed size is `4MB`) providing the findings and justification on the following topics:
  - Write a few lines about training dataset quality and any errors found in the training dataset.
  - Explain the data preprocessing steps.
  - Explain and justify the model you've chosen for calculating the index constituents.
- The source code of your approach for this task. Upload a *zip* file (maximum allowed size is `5MB`) with all relevant files to reproduce your results. The submitted file must have a `README` file with a detailed description about how to run the model to tag each of the raw job descriptions and to generate the `tags.tsv`. Do not forget to include links to any external libraries or packages you used for the generation of your model.

  There is no limit on execution time, but the code should generate the output file: `tags.tsv`.

Evaluation

Evaluation is done by calculating the F_1 score of your prediction. Let TP_i, FP_i, TN_i, and FN_i be the values of true positive, false positive, true negative, and false negative for the i^{th} $(1 \leq i \leq 12)$ tag. A tag assigned to a description is *positive* and a tag absent for a description is *negative*. We calculate the sum of true positives, false positives, true negatives, and false negatives over all the *twelve* tags:

$$STP = \sum_{i=1}^{12} TP_i$$

$$SFP = \sum_{i=1}^{12} FP_i$$

$$STN = \sum_{i=1}^{12} TN_i$$

$$SFN = \sum_{i=1}^{12} FN_i$$

Now we define precision P and recall R:

$$P = \frac{STP}{STP + SFP}$$

$$R = \frac{STP}{STP + SFN}$$

The accuracy of the prediction, or the F_1 score, is calculated as:

$$S = \frac{2PR}{P + R}$$

Your leaderboard score will be $10^3 \times S$.

Ranking

Prior to the end of the contest, the evaluation of your uploaded model's accuracy will be performed on a set of pre-selected 1446 descriptions from the `test.tsv`. At the end of the contest, your *last* uploaded file (i.e., the most recently uploaded file) will be used to calculate your final score and position on the leaderboard. Because of this, make sure that your final submission is the output file with the maximum score.

File Upload