**Mani Sundar Pathuri**

**Srinivas Rao PrithviNag Kolla**

**INFO 5731 Fall 2020**

**TERM PROJECT – FINAL REPORT**

Project name**: SEMI-SUPERVISED LEARNING FOR LEGAL DATA ARGUMENTATION**

**Abstract**

Now a day’s digital data is overflowing, Information is available everywhere and in vast amounts to people that it becomes difficult for an individual to understand in a short period of time.

We took up this Legal Research project as this fall into the above category. We see that Law professionals go through Large volumes of cases to have better grasp on past cases to take up future cases.

This helps them have a whole grip on the relevant previous cases to take better decisions during court sessions.

This is time consuming, so in order to fast track the process, NLP professionals in text classification parse huge documents of cases using machines to support. This will be a boost to law professionals in Legal Domain, which is generally seen down.

*Keywords: LEGAL dataset, Natural language processing, Semi-Supervised Learning*

**Introduction**

The legal domain is an under-explored area where machines have the potential to play an important role in alleviating the arduous tasks committed by law professionals by a daily basis. Previously work had been done on European Court of Human Rights data as legal corpora to classify argumentative and non-argumentative sentences in their argumentation structure(s). They were able to receive an 80% accuracy score in their binary classification task. Some Indian researchers have applied these methods on Indian Law. They used an annotator agreement system called Inter-Annotator Agreement (IAA), to judge the quality of the annotations, along with acknowledging the subjectivity existent in the labels. After the realization that hand-crafted features do not produce high results, the paper utilizes BiLSTM models, which achieves between 80-90% accuracy, for 7-class classification. It is possible to achieve similar results for U.S. law, using a similar process.

* To train the classifiers we took Labeled data set of published court reports with 6 component classes as: Invalid, Facts, Analysis, Conclusion, Issue, Rule/Law/Holding.
* 11 Train Data Sets, 1 Test Data Set, 1 Unlabeled Data Set.
* Columns & Type:

Text (String)

Target (String)

we aim to answer the following research questions:

**RQ1: How to handle the unbalanced data?**

**RQ2: Which type of algorithm is better, semi supervised or supervised?**

**RQ3: If supervised is better or semi supervised, then which classifier will be better?**

**Related works**

Compared to the numerous domains ML and NLP knowledge is applied to, the legal domain is an under-explored area where machines have the potential to play an important role in alleviating the arduous tasks committed by law professionals by a daily basis.

As argumentation is a crucial part of the overall project, explains what argumentation is and how to identify it in legal corpora. They use the European Court of Human Rights (ECHR) data as legal corpora to classify argumentative and non-argumentative sentences in their argumentation structure(s). They were able to receive an 80% accuracy score in their binary classification task.

Corpus annotation is a crucial step for the annotation task, as a principle in machine learning states “garbage in, garbage out”. understands to create an automated annotation system, manual annotation performed by 2 or more people is required, to ensure the quality of data is reliable for training. An important diagram in the paper states that for the highest of quality annotations, an annotator would require domain and linguistics knowledge and need to have a solid theory behind the annotation labels, to reduce ambiguity among annotators.

Potentially the most difficult phases to this research project are phases 3, 4, and 5, as they require a deeper study in the semantics and pragmatics of legal linguistics. A similar goal of implementation is thoroughly documented in, as their main focus is centered around “a ground truth for testing predictions about outcomes in new cases with new evidence; patterns for successful and unsuccessful argumentation; and guidance in retrieving, extracting, and organizing evidence for new arguments and new situations”. A creative approach that is taken for the construction of evidence-based arguments is a DeepQA architecture. This was the same architecture used to build IBM Watson for the game *Jeopardy*, except it is being used here to extract relevant arguments from past information using a relation extraction method.

In another part of the world, the same research project my team and I are researching is being conducted over in India by a group of Indian researchers. However, is applying similar principles and goals to Indian law, instead. They used an annotator agreement system called Inter-Annotator Agreement (IAA), to judge the quality of the annotations, along with acknowledging the subjectivity existent in the labels. After the realization that hand-crafted features do not produce high results, the paper utilizes BiLSTM models, which achieves between 80-90% accuracy, for 7-class classification. It is possible to achieve similar results for U.S. law, using a similar process implemented by this paper.

**Methodology**

When classes are imbalanced, standard classifiers are usually biased towards the majority class.

one category corresponds to methods that operate on the dataset in a preprocessing step preceding classification while a second category modifies the classification algorithm in order to put more emphasis on the minority class.

Second Category: cost-sensitive learning is a paradigm that emphasizes the incorrect classification of instances from the minority class during the training process while being minimal intrusive for the classifier in the sense that the algorithm does not require important changes.

Let’s understand imbalanced data with the help of an example.

**Example: Let us take a Fraudulent Data Set.**

Total Observations = 1000

Fraudulent Observations = 20

Non-Fraudulent Observations = 980

Event Rate= 2 %

Generally, Machine Learning Algorithms like Decision Tree and Logistic Regression have bias towards majority class data and often ignore and see Minority class data as Noice.

**Handling Un balanced Data**:

There are two methods by which we can handle the unbalanced data.

The first one is resampling techniques and the other is algorithmic ensemble techniques. Resampling technique means resample the original data to provide the balanced classes. There are 5 methods to resample the data. They are random under-sampling, random over sampling, cluster-based over sampling, synthetic minority over-sampling technique (SMOTE), and modified synthetic minority over-sampling technique (MSMOTE).

* 1. Resampling methods aim at modifying the dataset in order to reduce the discrepancy among the sizes of the classes.
  2. Random under sampling involves randomly selecting examples from the majority class and deleting them from the training dataset. In the random under-sampling, the majority class instances are discarded at random until a more balanced distribution is reached
  3. Random oversampling duplicates examples from the minority class in the training dataset and can result in overfitting for some models.
  4. This approach to addressing imbalanced data uses K-mean clustering. The clustering algorithm is applied to both the majority class and the minority class in which each class is oversampled, such that each class has the same number of data elements.
  5. SMOTE is Synthetic Minority Oversampling Technique. It works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.
     1. SMOTE is not effective for High Dimensional Data.
     2. The main drawback of oversampling Techniques is overfitting.

TF-IDF: Algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction.

**Supervised and Un-supervised method Application**:

To tell the better type of algorithm and the better classifier, we need to perform different types of algorithms on the modified data (after handling the unbalanced data) for both supervised and unsupervised. After performing the different types of algorithms both in supervised and unsupervised, we can tell the best one based on the performance metric (accuracy, confusion matrix, precision, recall, F score).

**Supervised**: All data is labeled, and the algorithms learn to predict the output from the input data.

Supervised learning problems can be further grouped into regression and classification problems.

* 1. **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
  2. **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

**Unsupervised**: All data is unlabeled, and the algorithms learn to inherent structure from the input data.

Unsupervised learning problems can be further grouped into clustering and association problems.

* 1. **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
  2. **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Classification Algorithms are:

* + - 1. Logistic Regression
      2. Decision Tree
      3. Random Forest
      4. Support Vector Machine
      5. Stochastic Gradient Descent

**Bert(Bidirectional Encoder Representations from Transformers)**:

* Bidirectional Encoder Representations from Transformers.
* The BERT algorithm is built on top of breakthrough techniques such as seq2seq (sequence-to-sequence) models and transformers. The seq2seq model is a network that converts a given sequence of words into a different sequence and is capable of relating the words that seem more important.
* BERT expects data in a specific format and the datasets are usually structured to have the following four features: guid: A unique id that represents an observation.

text\_a: The text we need to classify into given categories

text\_b: It is used when we’re training a model to understand the relationship between sentences and it

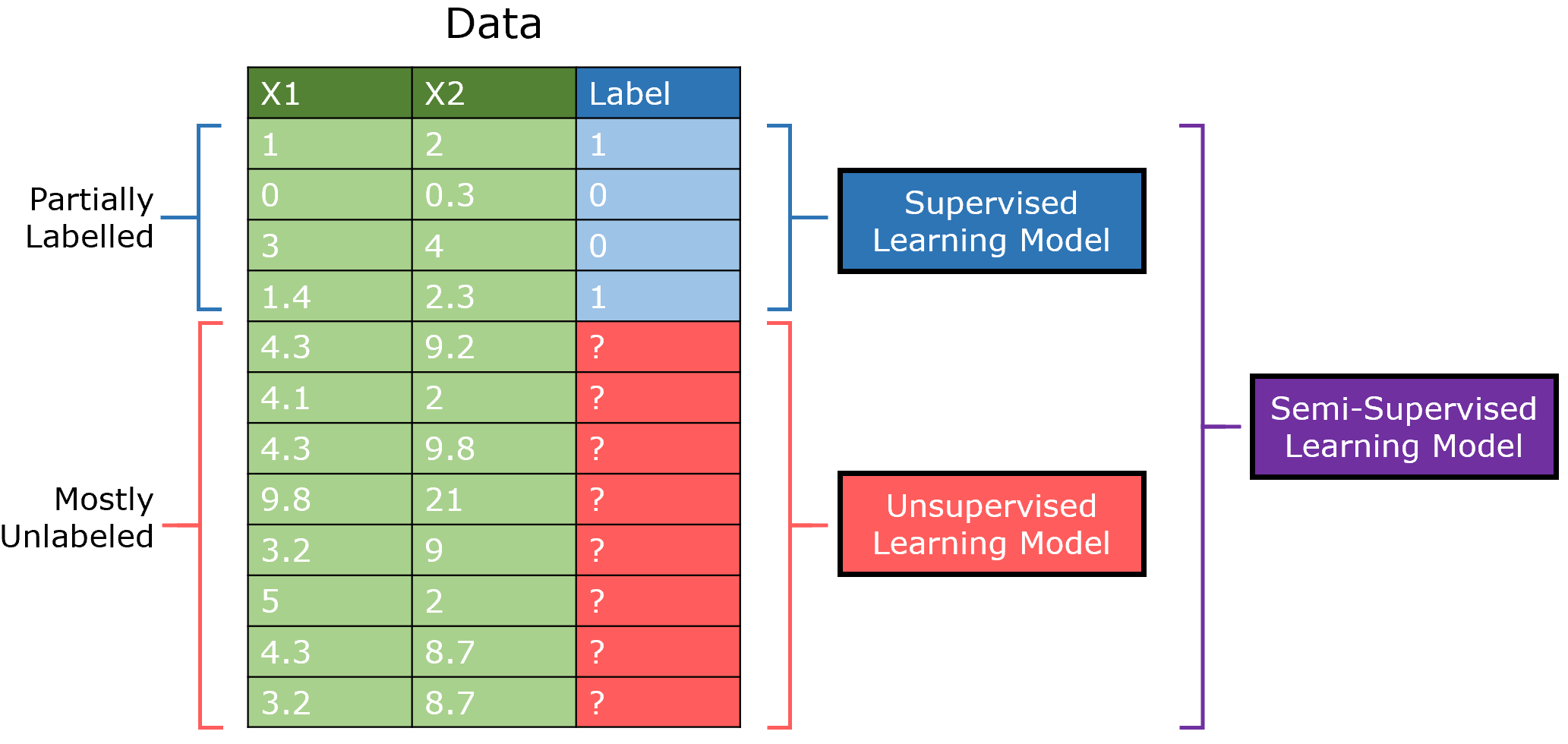
does not apply for classification problems.

label: It consists of the labels or classes or categories that a given text belongs to.

* The next step is pre-processing the data.
* Normalizing the text by converting all whitespace characters to spaces and casing the alphabets based on the type of model being used(Cased or Uncased).
* Tokenizing the text or splitting the sentence into words and splitting all punctuation characters from the text
* Adding CLS and SEP tokens to distinguish the beginning and the end of a sentence.
* Breaking words into Word Pieces based on similarity (i.e. “calling” -> [“call”, “##ing”])
* Mapping the words in the text to indexes using the BERT’s own vocabulary which is saved in BERT’s vocab.txt file.
* Consider the sentence ‘Oil prices extended its rally to a five-month high as conflict in Libya increased the risk of new supply outages.’
* Tokens : [‘oil’, ‘prices’, ‘extended’, ‘ its’, ‘ rally’, ‘to’, ‘a’, ‘ five’, ‘-’, ‘month’, ‘high’, ‘ as‘, ‘conflict’, ‘in’, ‘Libya’, increases’, ‘the’, ‘ risk’, ‘of’, ‘new’, ‘supply’, ‘out’, ‘##ages’.
* Input Ids: [101, 3534, 4563, 523,……]
* Input masks: [1,1,1,1,1,1,1,1,1….0,0]
* Segment IDs: [0,0,0,……….]
* The input IDs are token IDs with each ID representing a unique token. The input-masks help distinguish the tokens from the padding elements. In the above example, 0’s represent the padding elements.
* The padding is determined by the specified sequence length. If the length of the tokens is less than the specified sequence length, the tokenizer will perform padding to meet the sequence length.
* The segment IDs are used to distinguish different sentences. In the above example, we only have one text segment hence all Segment IDs are the same. If two sentences are to be processed, each word in the first sentence will be masked to 0 and each word in the second sentence will be masked to 1.

**Semi-Supervised Learning:**

* Take the same model that you used with your training set and that gave you good results.
* Use it now with your unlabeled test set to predict the outputs ( or pseudo-labels). We don’t know if these predictions are correct, but we do now have quite accurate labels and that’s what we aim in this step.
* Concatenate the training labels with the test set pseudo labels.
* Concatenate the features of the training set with the features of the test set.
* Finally, train the model in the same way you did before with the training set.



|  |  |
| --- | --- |
| **Data Set** | **Accuracy using smote and semi-supervised** |
| High kappa | 92.4 |
| Train 1 | 91.7 |
| Train 2 | 92.2 |
| Train 3 | 91.7 |
| Train 4 | 92 |
| Train 5 | 91.7 |
| Train 6 | 92.6 |
| Train 7 | 92.4 |
| Train 8 | 92.5 |
| Train 9 | 91.3 |
| Train 10 | 92.3 |

We also performed semi-supervised algorithm on the unbalanced High kappa dataset. We got an accuracy of 70%.

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Bert | 63.9 |
| SGB with semi supervised | 90.3 |

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| **MultinomialNB** | **59.2** |
| **Decision Tree** | **59** |
| **Random Forest** | **67** |
| **SGD** | **67.8** |
| **Logistic** | **65.8** |

|  |  |  |
| --- | --- | --- |
| **Data Set** | **Accuracy on training** | **Accuracy on testing** |
| **High kappa** | **70** | **60** |
| **Train 1** | **71** | **59** |
| **Train 2** | **70** | **63** |
| **Train 3** | **71** | **64** |
| **Train 4** | **70** | **61** |
| **Train 5** | **71** | **60** |
| **Train 6** | **71** | **64** |
| **Train 7** | **73** | **63** |
| **Train 8** | **71** | **62** |
| **Train 9** | **71** | **62** |
| **Train 10** | **70** | **61** |

**Conclusion**

In conclusion, we can say that semi supervised learning is better than supervised for text classification of multi labels. The unbalanced data set was handled by SMOTE. Although, we didn’t get good accuracy, semi-supervised learning helps to improve the accuracy by little margins which means more in the corporate world.

**Our project GitHub links:**

* + <https://github.com/sundarp17/sundar_info5731_fall2020/tree/master/project/train>
  + <https://github.com/sundarp17/sundar_info5731_fall2020/blob/master/semi_supervised.ipynb>
  + <https://github.com/sundarp17/sundar_info5731_fall2020/blob/master/handling_imbalance_data.ipynb>
  + <https://github.com/prithvikolla/Compuational_Methods_INFO5731/tree/master/Project_Legal_Text_Mining>

**References**

* Nair, A. (2019, November 18). Step By Step Guide To Implement Multi-Class Classification With BERT & TensorFlow. Retrieved December 11, 2020, from Analytics India Magazine website: <https://analyticsindiamag.com/step-by-step-guide-to-implement-multi-class-classification-with-bert-tensorflow/>
* Maxime. (2019, May 24). Introduction to semi-supervised learning and adversarial training. Retrieved December 11, 2020, from Medium website: [https://medium.com/inside-machine-learning/placeholder-3557ebb3d470](Maxime.%20(2019,%20May%2024).%20Introduction%20to%20semi-supervised%20learning%20and%20adversarial%20training.%20Retrieved%20December%2011,%202020,%20from%20Medium%20website:%20https:/medium.com/inside-machine-learning/placeholder-3557ebb3d470%20%20‌)
* ‌Cong, G., Lee, W. S., Wu, H., & Liu, B. (2004). Semi-supervised Text Classification Using Partitioned EM. *Database Systems for Advanced Applications*, 482–493. [https://doi.org/10.1007/978-3-540-24571-1\_45](Cong,%20G.,%20Lee,%20W.%20S.,%20Wu,%20H.,%20&%20Liu,%20B.%20(2004).%20Semi-supervised%20Text%20Classification%20Using%20Partitioned%20EM.%20Database%20Systems%20for%20Advanced%20Applications,%20482–493.%20https:/doi.org/10.1007/978-3-540-24571-1_45%20%20‌)
* ML | Semi-Supervised Learning. (2019, May 28). Retrieved December 11, 2020, from GeeksforGeeks website: [https://www.geeksforgeeks.org/ml-semi-supervised-learning/](ML%20|%20Semi-Supervised%20Learning.%20(2019,%20May%2028).%20Retrieved%20December%2011,%202020,%20from%20GeeksforGeeks%20website:%20https:/www.geeksforgeeks.org/ml-semi-supervised-learning/%20%20‌)

‌