Ensemble Approach to Causality Detection in Financial Text

Anonymous ACL submission

Abstract

Causality detection is one of the most fundamental and relevant natural language processing tasks. This task has a wide range of applications in various domains like health care, finance, and retail. Causality in sentences has variable syntax, making it challenging to identify and define using a single grammatical model. We address the problem of causality detection as a binary classification task of finding whether a sentence has causality or not. Financial causality detection focuses on discovering links between different assets from financial news to enhance trading strategies. This paper presents a comparative study using different machine learning algorithms on financial text to find causality. We have also implemented ensemble models to combine the results of weak learners. We achieved an F1 score of 93.4287 with the Extra Tree Classifier.

1 Introduction

Causality is an influence by which one event causes the formation of another event. The events in a natural language depend not only on how the events are mentioned implicitly or explicitly but also on the lexical representation of those events. "T causes S" or "S is caused by T" can be used to simply express causality in the text.

Financial news includes diverse accounts of financial objects' causes and consequences on sales, earnings, company decisions, and its economic effect on its stock prices. Causal analysis can be used to discover correlations between news and the values of various assets by identifying numerous repercussions (Qu and Kazakov, 2019). Causality identification can be used to predict stock movement and support financial services in the financial sector. As a result, even more financial analysis is required. We need to find the causal linkages between distinct sentences in the document to explain the variance in the data and make inferences. The purpose of Financial Causality Detection is to determine how a financial object or an event is modified based on a chain of events. The paper by (Kumar and Ravi, 2016) presented a survey of the text mining applications in the domain of finance. This paper covers some of the advancements of text mining algorithms in the field of finance. Predicting causality in the text is challenging because it is dependent on interactions between the text's author's communicative goals, the specific linguistic representation of information, the lexical context present in the text, and the causal relationships knowledge for the domain in focus.

This paper attempts to implement multiple machine learning algorithms using a simple TF-IDF weighting scheme for each sentence and still achieve an excellent F1 score which is comparable to F1 scores attained by deep learning models but require less time to train.

The following are the paper's key contributions:

- We used simple vector representation instead of complex vector representation to denote the sentences
- We experimented with various machine learning models and implemented ensemble learning models on top of them to improve upon the performance metrics
- The performance of these models was evaluated using Cochran's Q test to verify if there is a significant difference in their performance.

The rest of the paper is organized as follows: Section 2 discusses the literature review; Section 3 explains the dataset used in this paper; Section 4 covers the proposed methodologies; Section 5 illustrates the experimental results, and finally, Section 6 concludes the paper.

2 Related Work

In this section, various research concerned with causality detection is presented. A comprehensive survey on the automatic extraction of causal relations from the text is authored by (Asghar, 2016). This survey covered both statistical and non-statistical approaches to causal relation extraction.

A knowledge discovery system and knowledge extraction system that retrieves causal sentences from databases was designed (Khoo et al., 2000). The paper focused on identifying and extracting cause-and-effect connections found in medical datasets. The parse tree structure of the sentences that matched with the graph patterns was extracted as the causal relations. The model attained F1 score of 0.763 for 100 abstracts from the original four medical areas.

(Girju, 2003) developed an inductive learning approach that automatically detects and extracts causal relations from text. He tested the approach on the question answering system that gave precision 73.91% and recall 88.69%.

In the medical domain, to find causal relation between medication and virus mutations in the medical reports and journals (Bui et al., 2010) implemented the logistic regression model. The author was able to achieve F1 score of 84.5% using rule-based logistic regression for 500 sentences from PubMed abstracts.

(Blanco et al., 2008) used bagging, an ensemble machine learning model combined with the C4.5 decision trees technique, to detect, and extract causation. The algorithm had F1 score for causal sentences of 0.895 and non-causal sentences of 0.914.

A novel Restricted Hidden Naive Bayes learning method for dealing with feature interactions, such as causal connectives, and lexico-syntactic patterns, was presented by (Zhao et al., 2016). The model obtained F1 score of 85.6%.

SVM classifier and XGBoost model for causality detection in the financial sentences. In addition to this, a soft voting classifier was made by combining the SVM classifier and the XGBoost model (Pielka et al., 2020). The SVM classifier gave the best F1 score of 0.942, whereas the ensemble gave the best recall of 0.949 and precision of 0.950.

(Hariharan et al., 2020) built linear and deep learning models to classify whether a statement is causative or not. The SVM classifier outperformed the BERT uncased base model. The F1 score of SVM and fine tuned BERT model was 0.943 and 0.967 respectively.

3 Dataset

3.1 Dataset Description

We gathered the dataset from the organizers of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (Mariko et al., 2020). Quam collected the data from the corpus of financial news for the year 2019. The data is primarily provided in the form of CSV files. The dataset has the following headers: Index (Identification number for the passage), Text (Passages collected from the news), and Gold (The result label obtained by manual annotation). The data is separated into two categories based on whether the underlying structure of sentence does have causality (causal sentence) and does not have causality (non-causal sentence). The gold header denotes whether the passage is causal or not; for example, a passage labeled 1 is causal, whereas a passage labeled 0 is not. The dataset consists of 22,058 sentences. The average size of the sentences in the dataset is about 214 characters. The dataset is largely imbalanced as the ratio of non-causal sentences to causal sentences is 13:1 approximately, i.e., for every causal sentence, there are 13 non-causal sentences.

3.2 Preprocessing

The main motivation behind the preprocessing is to eliminate the uninformative words such as stop words, numbers, currency symbols, time specifications like “am” and “pm,” dates, abbreviations, punctuations, URLs, and HTML tags. Here we assume the uninformative words are the ones that don’t communicate any information about the sentence’s categorization, i.e., whether it’s causal or not. This preprocessing phase is based on the idea that words with identical numerical parameters, such as the amount of money and dates, can be categorized into various classes, which is not desirable. We further used lemmatization to stem the words into the root words.

4 Proposed Methodologies

4.1 Feature representation

The sentences are denoted using the Term frequency-inverse document frequency (TF-IDF)

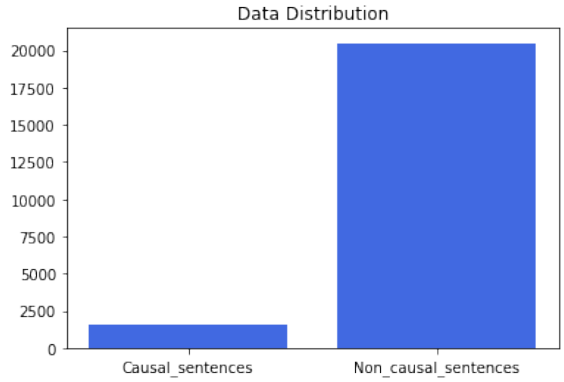

Figure 1: Dataset Statistics

Type	No of sentences	Percentage
Causal sentences	1,579	7.2
Non-causal sentences	20,479	92.8

Table 1: Dataset distribution.

representation. The TF-IDF(Salton and McGill, 1983) is intended to show how relevant a term is in a given document. The TF-IDF vectorizer was chosen because it works by proportionately increasing the number of times a word appears in a document while balancing it out with the number of documents in which the word appears. For the TF-IDF vectorizer, the parameters used are min-df, max-df, and ngrams. We have set max-df (Maximum Document Frequency) = 0.2 implies that it ignores the terms that are seen in more than 20% of the documents, whereas min-df (Minimum Document Frequency) = 3 implies that it ignores the terms that occur in less than three documents. The ngram range used was (1, 3).

4.2 Machine Learning Architecture

Several machine learning-based standard classifiers are used for causality detection. Finally, the outputs of these classifiers are combined using ensemble techniques to improve the accuracy of the predictions.

4.2.1 Logistic Regression

In the case of classification problems, logistic regression is widely implemented. It is used when the value of the output has categorical values. The probability for classification problems with two possible outcomes is modeled using logistic regression. Logistic Regression(Genkin et al., 2007) gives good results in the case of binary classification problems.

4.2.2 Linear SVM

SVM is a machine learning classification technique that was introduced and is considered effective. The main idea of this algorithm is that it generates a boundary line or a hyperplane which is used to split up the data into classes. When there are a lot of features, the linear kernel is preferred for text classification since it performs well(Tong and Koller, 2001). Hence, SVM linear model has been used in this paper.

4.2.3 Multinomial Naive Bayes

The multinomial Naive Bayes algorithm is based on the Bayes theorem. MNB (McCallum et al., 1998) is the method of learning probability that is commonly used in Natural Language Processing(NLP). Since it is a probabilistic classifier, it computes the conditional probability of each category given a text using the Bayes theorem and then outputs the category with the highest conditional probability.

4.3 Ensemble Learning

Ensemble learning is a machine learning approach that integrates predictions from many models to improve overall predictive performance. When compared to a single model, this approach provides greater predictive performance. Despite the fact that there appears to be no limit to the number of ensembles one can generate for your predictive modeling, three approaches dominate the field of ensemble learning: bagging, boosting, and stacking.

4.3.1 Voting Ensemble

Voting Ensemble simply aggregates the results of each classifier supplied into the Voting Classifier and predicts the output class with the highest votes. The objective behind the voting ensemble is to create a single model that trains on several models and predicts output based on the cumulative majority of votes for each output class, rather than building numerous models and determining their precision. The voting classifier has two primary hyperparameters: estimators and voting. The voting hyperparameter may be adjusted to either hard or soft, and the estimator's hyperparameter is simply classifiers that are utilized in the ensemble. Hard voting involves summing up each class label's predictions

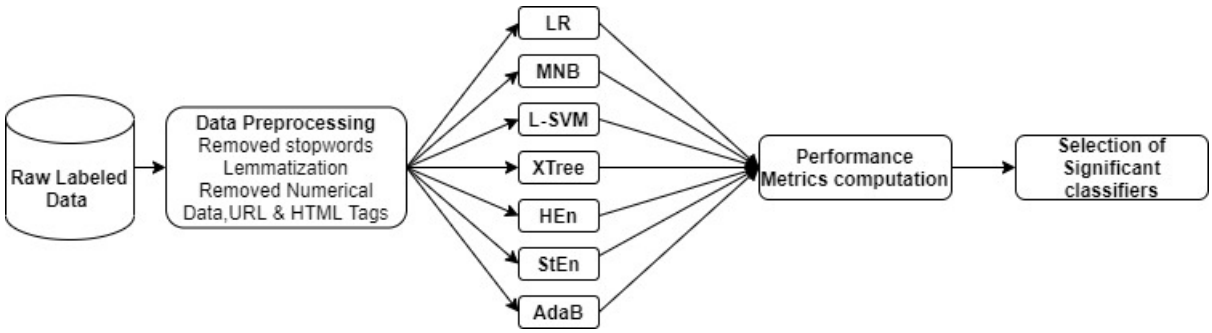

Figure 2: Framework used in the Experiment

and predicting the class label with the most votes. Soft voting involves combining all of the predicted probabilities for each class label and predicting the one with the highest likelihood. In this paper, we use hard voting as it is appropriate in predicting the distinct class labels, which is the case here.

4.3.2 Extra Trees Classifier

Extra Trees Classifier is an ensemble machine learning algorithm based on decision trees. The Extra Tree Classifier functions by using the training dataset to generate a large number of unpruned decision trees. Predictions are formed by averaging the decision tree predictions in the case of regression, and majority voting is utilized based on the decision tree predictions in the case of classification tasks. As our task is the classification, we use a major voting approach. Extra Trees classifier fits each decision tree to the entire training dataset, unlike bagging and random forest, which create each decision tree using a bootstrap sample of the training dataset.

4.3.3 Stacking

Stacking is one of the ensemble learning methods. The main idea behind stacking is to search for a varied collection of members by altering the model types to fit the training data and employing a model to combine predictions. The stacking architecture involves base models (level-0 models) which fit on the training data and whose predictions are compiled, and a meta-model (level-1 model) that combines the predictions of all the base models. Although stacking is designed to increase modeling performance, it is not always guaranteed to do so in all circumstances.

4.3.4 AdaBoost

AdaBoost(Adaptive Boosting) is a type of ensemble learning method designed to improve binary classifiers' efficiency. The idea behind AdaBoost is that while a single classifier might not be able to predict an object's class accurately, we can build a powerful model by aggregating several weak classifiers, each learning from the mistakenly classified items of the others. One-level decision trees, also known as decision stumps, are the most suitable and hence most commonly employed approach with AdaBoost since the trees are small and only have one classification decision.

5 Experiment and Results

5.1 Evaluation

Since there are no distinct datasets, such as the training and testing dataset, hence the dataset is split in such a way that training data makes up 80% of the dataset, whereas the testing data makes up 20% of the dataset. This split is done at random, and sentences in the training and testing data are not repeated.

5.2 Training the model

We implemented seven machine learning models for the binary classification task at hand. Traditional classifiers included Logistic Regression (LR), Linear SVM (L-SVM), and Multinomial Naive Bayes (MNB). Ensemble classifiers included AdaBoost (AdaB), Extra Tree Classifier (XTree), Hard Voting Ensemble Classifier(HEn), and Stacking Ensemble Classifier(StEn).

Hard Voting Ensemble (HEn) consists of the following estimators- Logistic Regression (LR), Linear SVM (L-SVM), Multinomial Naive Bayes (MNB), and Extra Tree Classifier (XTree).

Stacking Ensemble(StEn) uses Logistic Regression (LR), Multinomial Naive Bayes (MNB), and Extra Tree Classifier (XTree) as base models (level-0 models) and Linear SVM (L-SVM) as the meta-model (level-1 model).

Model	Parameters
Logistic Regression	penalty = l2; solver = liblinear; multi_class = ovr
Linear SVM	Regularization parameter, C = 1.0; kernel = linear, class_weight = balanced
Multinomial Naive Bayes	alpha = 1.0; fit_prior = True
ExtraTree Classifier	n_estimators=100, criterion = "gini"
Ada Boost	n_estimators=30, random_state=10
Stacking	estimators = (LR, MNB,XTree); final_estimator = L-SVM
Voting Ensemble	estimators = (LR, MNB, XTree, L-SVM); voting = 'hard'

Table 2: Optimised set of parameters for ML models

5.3 Results

Table 3 shows the results obtained from our experiments on the dataset with the traditional classifiers and ensemble classifiers. The metrics which we computed are Accuracy, Precision, F1-Score, Recall. Based on the performance measures, the following observations may be drawn from Table 3-

- In general, ensemble classifiers performed better than individual classifiers.
- The Extra Tree Classifier (XTree) achieved the best results with its accuracy and F1-score being relatively more than other classifiers.
- Linear SVM (L-SVM) performed the best among the traditional classifiers, with its F1-score being second only to Extra Tree Classifier (XTree).

5.4 Statistical Test

Cochran's Q test(Tate and Brown, 1970) is used for Hypothesis Testing. The Cochran's Q test is an extended form of McNemar's test that may be used to evaluate several classifiers at the same time. Cochran's Q test formally tests the hypothesis that the classification accuracies of the models are the same or not. We have used Cochran's Q test as it can be used to compare more than two classifiers which is the case here.

After the performance metrics of different models were obtained, Cochran's Q test was applied to find the significant classifiers. The null hypothesis H0: There is no significant difference between the classification accuracies of models Logistic Regression (LR), Linear SVM (L-SVM), Multinomial Naive Bayes (MNB), AdaBoost (AdaB), Extra Tree Classifier (XTree), Hard Voting Ensemble Classifier(HEn) and Stacking Ensemble Classifier(StEn).

The value of the significance level α was chosen as 0.05. After applying Cochran's Q, we got the p-value as 0.01767, which is less than the significance level α. Thus, the null hypothesis is rejected, which implies a significant difference in the classification accuracy of the models. Since the null hypothesis is successfully rejected, multiple pairwise post hoc pairwise tests are performed to determine which pairs of models have significant differences in their classification.

In Figure 3, the green arrow signifies there is no significant difference between the classification accuracies of the different models, whereas the red cross signifies there is a significant difference between the classification accuracies of the different models.

The models Linear SVM (L-SVM), Extra Tree Classifier (XTree), Hard Voting Ensemble Classifier(HEn), and Stacking Ensemble Classifier(StEn) do not have a significant difference between their classification accuracies.

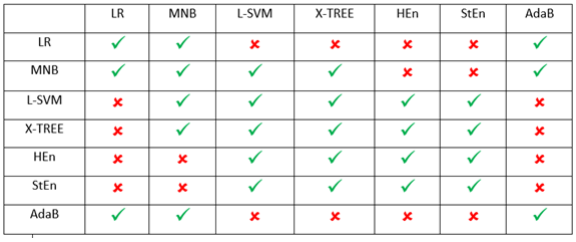

Figure 3: Comparison of classification accuracies of different classification models

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.942198	0.93436	0.939937	0.924459
Linear SVM	0.943694	0.93885	0.943790	0.931437
Multinomial Naive Bayes	0.942923	0.939394	0.941976	0.926719
ExtraTree Classifier	0.944827	0.940519	0.945376	0.934287
Ada Boost	0.935533	0.927020	0.937670	0.926789
Stacking	0.944147	0.943439	0.944243	0.929972
Voting Ensemble	0.943921	0.944232	0.943563	0.928219

Table 3: Overall metrics for various classifiers.

6 Conclusion and Future Work

In this paper, we have implemented seven classifiers for the binary classification of financial textual data to detect whether they have a causality relation or not. For feature representation of financial sentences, we have used TF-IDF vectorizer in our approach. Since the classification accuracies of the seven models were approximately very close, we applied the Cochran's Q test to check if there was a significant difference between the classification accuracies of the seven models or not. However, the null hypothesis was rejected, which implied that there was indeed a significant difference between the classification accuracies of the seven models. Among the seven classifiers, the top 3 classifiers for this dataset based on accuracy value are Extra Trees Classifier, Stacking, Voting Ensemble, but it was discovered that the accuracy of the top three classifiers was not significantly different after performing the multiple pairwise post hoc pairwise tests.

Further steps would be to improve the results by using the oversampling techniques to oversample the causal sentences as the dataset is skewed toward non-causal sentences. In addition to this, we intend to experiment with feature selection on the vectorized data to see its effect on the results.

References

Nabiha Asghar. 2016. Automatic extraction of causal relations from natural language texts: a comprehensive survey. *arXiv preprint arXiv:1605.07895*.

Eduardo Blanco, Nuria Castell, and Dan Moldovan. 2008. Causal relation extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Quoc-Chinh Bui, Breanndán Ó Nualláin, Charles A Boucher, and Peter MA Sloot. 2010. Extracting causal relations on hiv drug resistance from literature. *BMC bioinformatics*, 11(1):1–11.

Alexander Genkin, David D Lewis, and David Madigan. 2007. Large-scale bayesian logistic regression for text categorization. *technometrics*, 49(3):291–304.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, page 76–83, USA. Association for Computational Linguistics.

RL Hariharan et al. 2020. Nitk nlp at fincausal-2020 task 1 using bert and linear models. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 60–63.

Christopher SG Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 336–343.

B Shravan Kumar and Vadlamani Ravi. 2016. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114:128–147.

Dominique Mariko, Hanna Abi Akl, Estelle Labidurie, Stephane Durfort, Hugues De Mazancourt, and Mahmoud El-Haj. 2020. Financial document causality detection shared task (fincausal 2020). *arXiv preprint arXiv:2012.02505*.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Maren Pielka, Rajkumar Ramamurthy, Anna Ladi, Eduardo Brito, Clayton Chapman, Paul Mayer, and Rafet Sifa. 2020. Fraunhofer iais at fincausal 2020, tasks 1 & 2: Using ensemble methods and sequence tagging to detect causality in financial documents. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 64–68.

Haizhou Qu and Dimitar Kazakov. 2019. Detecting causal links between financial news and stocks.

In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–8. IEEE.

Gerard Salton and Michael J McGill. 1983. *Introduction to modern information retrieval*. mcgraw-hill.

Merle W Tate and Sara M Brown. 1970. Note on the cochran q test. *Journal of the American Statistical Association*, 65(329):155–160.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Sendong Zhao, Ting Liu, Sicheng Zhao, Yiheng Chen, and Jian-Yun Nie. 2016. Event causality extraction based on connectives analysis. *Neurocomputing*, 173:1943–1950.