Anuj@FINSIM–Learning Semantic Representation of Financial Domain with Investopedia

Anuj Saini
Publicis Sapient, Los Angeles, CA, USA
anuj.saini@publicissapient.com

Abstract

Natural Language Processing and its applications are getting used in every domain, and it has become an important need to have domain specific knowledge representation in the form of ontologies, taxonomies or word embeddings like BERT. As most of these knowledge bases are generic and lack the specificity of a domain, it is very important to have semantic representation for domain separately. The FinSim 2020 shared task is colocated with the FinNLP workshop, and the challenge is to classify financial terms into their predefined classes or hypernyms. This paper explains a hybrid approach that uses various NLP, machine learning, and deep learning models to develop a financial terms classifier. Also the paper explains use of a financial domain encyclopedia called Investopedia to enrich terms for better context. The semantic representation of financial terms is a very important building block for NLP applications such as question answering, chatbot, trading applications etc.

Keywords
Financial Ontology, BERT, Investopedia, Machine Learning, Natural Language Processing, Support Vector Machine, Word Embeddings, Ontology

1 Introduction

Knowledge is semantic representation of data and can be defined in various forms such as ontologies with entity-relations or taxonomies with hypernyms/hyponyms relations or in the form of word embedding. Semantic knowledge representation is the core building block task of NLP systems and has been there since decades. A lot of work has been done on systems like OpenCyc, FreeBase, YAGO, DbPedia etc. However, most of these systems are generic and lack specialized detailed terms and entities such as medicine names for healthcare or financial lingo for financial domain. Another issue is that earlier work in the domain of semantic representation is mostly manual and took years of efforts. With technology and AI advancements along with high computational power available, a lot of work has been undergoing to develop various language models. Language models represent knowledge in the form of embedding or vectors which are context aware and can be used for various applications such as text similarity, machine translation or word prediction. Systems like BERT have stormed the NLP space and are beating most benchmarks across all NLP applications. XLNet, RoBERTa, ELMO etc. are some other different kind of word embeddings which are pretrained and can be easily applied for different NLP applications.

A lot of work has been done to develop financial domain specific knowledge base and embeddings. FIBO (Financial Industry Business Ontology) [Bennett, M. 2013] is an owl representation of entities and their relations. Similarity FinBERT [Araci, D. 2019] is BERT [Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018)] model trained on a huge financial corpus and provide pre-trained model for financial domain tasks.

The paper has been organized as follows. Section 2 gives the description of the training dataset provided by the FinSim organizing committee. Section 3 presents the proposed approach for Financial Terms Classification prediction. The experimental evaluation has been carried out in Section 4. Section 5 concludes our research work followed by acknowledgment and references given in Section 6 and Section 7 respectively.

2 Related Work

Similar tasks have been carried across different levels but most of them were generic in nature. SemEval-2015 [Georgeta Bordea, Paul Buitelaar, Stefano Faralli and Roberto Navigli (2015)] asked participants to find hypernym-hyponym relations between given terms. Similar work to extract knowledge from unstructured task were done at TAC, where the task was to develop and evaluate technologies for populating knowledge bases (KBs) from unstructured text.

3 Data Set Description

FINSIM is a supervised financial term classification task. The task has a total of 100 data points with 2 column data having term to be classified and its class. There are in total 8 classes (hypernyms) and training data was also imbalance wherein classes such as Bonds having 28% data and class Forward has only 5 data points. Overall, 100 data points is too little to train a model, so definitely we needed some more context to enrich data for better performance. Below is the distribution of classes and their respective counts from the training data as shown in table 1. For test data, there were 99 terms to be classified into the correct hypernym.

Data has only two columns: terms and their classes they belongs too. So there was not enough context for text to classify it into its right hypernym or class. There are terms which are self explanatory like Corporate Bond or Equity Future but on the other hand a lot of abbreviations have been used specially in test data like TIPS or ABS which are totally impossible to classify without having enough context about these terms. That is where we tried to use external embeddings but eventually settled with Investopedia which is a rich dictionary of Financial terms. We enriched our data using Investopedia definitions and other Investopedia topics containing term to classify into it.

Class	Count
Bonds	28
Swap	18
Option	12
Funds	11
Future	9
MMIs	9
Stocks	8
Forward	5

Table 1: Classes Count

3 The Proposed Solution

The proposed approach includes enriching text with Investopedia as first step so that we have enough context for classification. Further, we carried out standard text preprocessing steps and then feature engineering, which includes a set of new features that we generated out of raw text and then trained model using various classification algorithms.

3.1 Investopedia

Investopedia is the world's leading source of financial content on the web which contains a huge financial dictionary for all financial jargons. We collected in total of 15347 financial terms and their definition and detailed description out of Investopedia by simply scrapping site https://www.investopedia.com/. Then we created definition of all the terms for training and test data. This enrichment of texts really helped a lot especially for financial terms which had very little context or were complete slangs or abbreviations. Some examples of Investopedia enrichment are:

Example *TIPS: Treasury inflation-protected securities (TIPS), The actual financial benefit of an investment after accounting for inflation and taxes. The after-tax real rate of return is an accurate measure of investment earnings and usually differs significantly from an investment's nominal rate of return,…*
Example *T-note:The purchase of treasury notes or bonds from dealers, by the Federal Reserve. The "coupon" refers to the coupons which are the main difference between T-notes and T-bills….*

So after the data enrichment process, we had a full definition of terms rather than just the terms for both training and test data.

3.2 Text Pre-Processing

For each data point, the preprocessing steps are as follows:

Text Encoding
We have encoded the definitions using standard UTF-8 encoding that handles scripting of foreign languages.

Tokenization
We applied text tokenization for data analysis and finding important words or tokens, also for removing punctuations and stop words we needed tokenization. We have tokenized the financial text into words using NLTK library.

Punctuation and numbers extraction
All numbers and punctuations are simply don't help in hypernym classification of text. So we used regexes to extract functions and digits.

Stop Words Filtering
We have used standard English stopwords to extract and mark them as English (en) language

WordNet
We have used English WordNet to detect synonyms for terms in general English. WordNet is an extensive lexical dictionary mostly being crafted manually and English WordNet has around 200K word and can be downloaded from the mentioned link in [5]. We used English WordNet form the NLTK python package.

POS Tagging
Part of Speech tagging is the process of tagging every token with its grammatical tag such as noun/adverbs/adjective etc.

As we are more focused on nouns here, it is important to label each word in order to identify nouns.

3.3 Feature Engineering

We tried a lot of various features and tested on different models. We list below all the features created and tested models on:

Character Count
Simply taking count of term in terms of characters.

Words Count
Number of words in the term to be classified.

Word Density
Computed as character Count/words count.

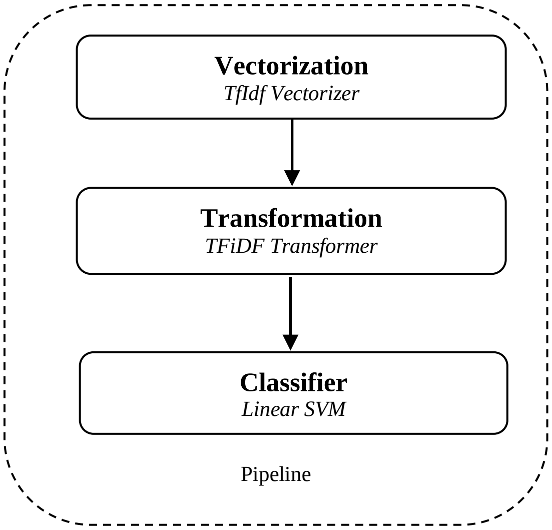
Title Words Count
Words starting with upper case letter.

Upper Case Words Count
Number of words where all characters are upper case letters.

Nouns Count
Number of nouns in the text.

Adjectives Count
Number of adjectives in the text.

Words Embedding
We also tried to use wiki-news-300d-1M.vec embeddings and generated token level embeddings.



5 Model and Pipeline

We experimented a number of models along with text processing pipeline including CountVectorizer and TfIdfVectorizer. We created a NLP pipeline which pre-processes raw text and then step by step transforms text into vectors, run model, and generates results using a SVM classifier.

5.1 Vectorization

Vectorization is an important process of converting text into numbers. We tried a number of approaches here starting with CountVectorizers that simply count the occurrences of words in the text, CountVectorizers at character level with trigrams. Finally, we got best results using TfIdf Vectorizer with the following configurations:

Analyzer
Splitting text by set of characters or words. We used word level, as we had enough text to get tfidf scores at word level.

Ngram-range
This feature creates ngrams for the given text. We got good results with bigrams.

Max Features
Property to define how many total words to be used for the model to train on. With ngrams enabled and text enriched using Investopedia, we had a lot irrelevant tokens to consume. So we limit our model to use only top 5,000 tokens.

5.2 Transformation

The next step in our pipeline is to transform text into vectors defined in the previous step. We used TfidfTransformer to convert text into vectors.

5.3 Classification

The last step in our pipeline is to classify text into hypernyms. Again, here we attempted different models like RandomForest, XGBoost, Naïve Bayes etc. We got better results with Support Vector Machine (SVM) using LinearSVC with the following configuration:

Kernel
We tried various kernel and cleaner data and features, we got better results with linear kernel.

Penalty
Penalty specifies the norm used in the penalization. We used standard SVC penalty, which is ‘l2’ penalty.

5.4 Multi-label Classification

One of the ask in this task was to generate all labels in ranked order. Therefore, we have to train our model with probability=true parameter in LinearSVC model that results in probability for each class.
Example (Green Bond): *[('Bonds', 0.5137775373484725), ('MMIs', 0.12466465491304607), ('Swap', 0.07597307049335765), ('Funds', 0.06947138280959159), ('Future', 0.06498779183269941), ('Option', 0.061816001759789255), ('Stocks',*

0.052482381962057296), ('Forward',
0.03682717888098626)]

6 Results

As there were only 100 data points to develop and test models and even certain classes had too few occurrences, so we decided to use 28 cross validation cycles. Cross-Validation also known as CV is a better way to test data specially when we don’t have enough data to train as well test on. Given formula to compute Cross Validation.

$$\boldsymbol{CV(\lambda) = 1/K \sum(K, k=1)\, Ek(\lambda)}$$

Where K=28

Still the official results were not the results that we were expecting. The final task was to generate multi-label classification, but for benchmarking our results we restricted the model to validate data on single class only. Finally, after trying various models and vectorization we got our best model results mentioned below in table 2.

class	**precision**	**recall**	**f1-score**	**support**
Bonds	0.74	0.89	0.81	28
Forward	1	0.6	0.75	5
Funds	1	0.82	0.9	11
Future	1	0.89	0.94	9
MMIs	0.71	0.56	0.63	9
Option	0.86	1	0.92	12
Stocks	0.88	0.88	0.88	8
Swap	1	0.94	0.97	18

Table 2: CV Scores Summary

With overall accuracy of 87% and average precision of 89% is what we got best from our cross validation tests.

6 Future Work

Building fully automatic or semi-automatic taxonomies or ontologies is an important NLP task. This shared task is the perfect step in the direction of developing taxonomies in the financial domain. There are a lot of other experiments that can add more accuracy to the solution. The most important missing method is using BERT or FinBERT embeddings. As BERT is trained on a huge corpus so it can definitely bring a lot of context to the solution, it can help to find out relationships between hypernyms and the term. Secondly, organizers have provided FIBO ontology as part of resources, but we haven’t used that and it could definitely have helped us to get more information related to each term. FIBO ontology has triples for entities and it can be useful to identify the full form of abbreviations or even direct hypernyms mentioned in the ontology.

Another important observation and definitely a need to be extended as part of this research is to add more training data to the corpus. It will help the community to train better and robust model with more experiments. This research can be further extended toward building an automatic ontology wherein we can extract triples, find out relationship (synonym, hypernym, hyponym etc.) between term and class, or classifying a term into its type like company, property or concept etc.

7 Conclusion

Natural Language Processing has a very big role to play especially in the financial domain. A lot of financial reports needs to be parsed and consumed which require a lot semantic text understanding. Having a semantic knowledge base is the most important building block for any text parsing system. Language models and word embeddings have their own limitations as they do not tell us much about the quality of the results, no explainability or for instance won’t give us the type of relation. So our understanding is that a hybrid solution for this task will be a way forward. Moreover, most existing work is highly manual and stale at current state whereas the financial world is getting daily updated with new companies or new financial jargons.

We have tried to keep our solution very simple and standard with a simple pipeline and focused more on data analysis and data enrichment. We simply used common machine learning instead of making more complex models using deep learning or BERT embeddings simply because of lack of lot of training instances.

Acknowledgements

We would like to thank organizers [Ismail El Maarouf and Youness Mansar and Virginie Mouilleron and Dialekti Valsamou-Stanislawski 2020] for conducting this shared task and also building the training data. We also would like to thank Sapient Corporation for giving us an opportunity to work and explore the world of text analytics.

References

[Araci, D. 2019] Finbert: Financial sentiment analysis with pre-trained language models. Computing Research Repository, arXiv:1908.10063

[Georgeta Bordea, Paul Buitelaar, Stefano Faralli and Roberto Navigli (2015)] “SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval)”. In *Proceedings of SemEval 2015*, co-located with NAACL HLT 2015, Denver, Col, USA.

[Georgeta Bordea, Els Lefever, and Paul Buitelaar (2016). “Semeval-2016 task 13] Taxonomy extraction evaluation (TExEval-2)”. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, CA, USA.

[Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion (2018)] “SemEval-2018 Task 9: Hypernym Discovery”. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, United States. Association for Computational Linguistics.

[Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018)] “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. https://arxiv.org/abs/1810.04805v2.

[Bennett, M. 2013] The financial industry business ontology: Best practice for big data. J Bank Regul 14, 255–268. https://doi.org/10.1057/jbr.2013.13

[Ismail El~Maarouf and Youness Mansar and Virginie Mouilleron and Dialekti Valsamou-Stanislawski 2020] "The FinSim 2020 Shared Task: Learning Semantic Representations for the Financial Domain",proceedings of IJCAI-PRICAI 2020"