FINSIM20 at the FinSim Task: Making Sense of Text in Financial Domain

Vivek Anand[1*] , **Yash Agrawal**[1*] , **Aarti Pol**[2] and **Vasudeva Varma**[1]
[1]International Institute of Information Technology, Hyderabad
[2]VIT, Pune
{vivek.a, yash.agrawal}@research.iiit.ac.in, aarti.pol12@vit.edu, vv@iiit.ac.in

Abstract

Semantics play an important role when it comes to automated systems using text or language and it is different for different domains. In this paper, we tackle the FinSim 2020 shared task at IJCAI-PRICAI 2020. The task deals with designing a semantic model which can automatically classify short phrases/terms from financial domain into the most relevant hypernym (or top-level) concept in an external ontology. We perform several experiments using different kinds of word and phrase level embeddings to solve the problem in an unsupervised manner. We also explore the use of a supplementary financial domain data; either to learn better concept representation or generate more training samples. We discuss both the positive and negative results that we observed while applying these approaches.

1 Introduction

Semantics has been a tough area in NLP research. This also comes in the disguise of getting to know the taxonomy or hypernymy relations of terms. There have been tasks in NLP for this purpose [Bordea *et al.*, 2016; Camacho-Collados *et al.*, 2018]. These tasks get tougher when applied for a specific domain; for example, the word ”stocks” can have several meaning in general sense but while in financial domain; the meaning can be narrowed down. Thus making the semantics a bit more clear.

The purpose of FinSim 2020 Shared Task is to automatically map financial phrases/terms to a more general financial concept. Alternatively, it means to map a hyponymy to its hypernymy. This kind of task in financial domain has been introduced for the first time.

The task provides us with a training data that maps some of the financial concepts to its hypernymy; for example, ”Alternative Debenture” is mapped to ”Bonds”. The given set of hypernymy labels has cardinality of 8. This set includes Bonds, Forward, Funds, Future, MMIs, Option, Stocks and Swaps. The pre-mapped data (training data) contains very low number of labelled examples so we explore unsupervised techniques.

We explore the use of pre-trained word embeddings [Pennington *et al.*, 2014]. These pre-trained word embeddings are trained on general corpus and not domain specific. We make use of the given training data to explore if the pre-trained word embeddings can be used for this financial domain task. We perform experiments with several kinds of unsupervised algorithms based on word embeddings using cosine similarity and variants of KNN algorithm as well as using some deep learning based methods.

Further the task also provides us with corpus of financial domain text. This text can be used to learn representations or patterns useful for the task. We explore the use of Hearst Patterns [Hearst, 1992; Seitner *et al.*, 2016] on this text to automatically mine hypernymy-hyponymy relations from the given text which is useful in extending the training dataset. We perform similar experiments on this extended dataset and report the results.

The remainder of paper is organized as follows. Section 2 describes the approaches we have tried for the task. Section 3 describes experimental setting and some details regarding the approaches. Section 4 describes the results achieved from several methods and out ranking in the task. We then conclude in Section 5.

2 Our Approach

2.1 Cosine Similarity Based

We explore the use of pre-trained word and sentence embeddings in this approach. For basleine, we consider GloVe [Pennington *et al.*, 2014] word embeddings and it's finetuned versioned on given financial documents. Since, GloVe embedding are word based, to get the embedding of financial concept; we take pre-trained word embedding of each word in the input financial concept and average it. For each of the input financial concept we try and map it to its hypernymy using the averaged word of both the input financial concept and the hypernymy label itself. We find the cosine similarity with each of the average embedding of hypernymy labels to get a ranked list of labels for a given financial concept.

We also experiment with mapping financial concept to averaged embedding of the description of hypernymy as per The Financial Industry Business Ontology (FIBO). Due to the av-

[*] Authors contributed equally.

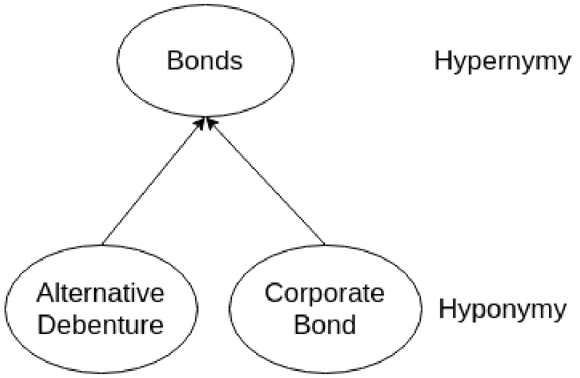

Figure 1: Example of financial concept labeling

eraging, description based approach performed poorly compared to just using embedding of hypernymy. For sentence embeddings, we use Universal Sentence Encoder (USE) [Cer *et al.*, 2018] pre-trained embeddings. USE gives 512 dimensional embedding of the given input phrase. A t-SNE representation of these embedding is for each financial concept is shown in Figure 3. Again, we find the cosine similarity of given financial concept with each of the hypernymy labels to get a ranked list of labels for that concept. In case of USE, description of labels performs better than GloVe embeddings but is still worse than using hypernymy labels only.

2.2 Deep Learning Based

We have very few labeltavled examples to train a supervised model and the class distribution isn't consistent. For example, label "bonds" and "swap" have close to half training samples and other half is distributed among six labels. In order to handle class imbalance, we use weighted cross-entropy loss function. We experiment with CNN for text [Kim, 2014], LSTM [Hochreiter and Schmidhuber, 1997] and a transformer based RoBERTa model [Liu *et al.*, 2019]. We find that these supervised model are able to learn the task to a reasonably good extent and all gives almost similar results. However, even with a very large number of parameters, RoBERTa model gives only comparable scores to both CNN and LSTM based models. This might be due to the fact that RoBERTa model need larger number of labelled samples for the finetuning.

2.3 Naive KNN Based

In this approach, the main idea is to map the given input financial concept with one of the financial concept in the training set. We use average pre-trained embedding to get representation of the given financial concept. We get the cosine similarity score with average pre-trained embedding of all the financial concepts present in the training set. We only consider the top k most similar financial concepts to the input financial concept. We finally consider the label of these k financial concepts and output the most frequent label.

This intuition behind this approach is that the input financial concept will be most similar to all the other financial concepts which come under the same category. Alternatively, it can be said as find the most similar sibling and concluding they have same parent.

2.4 Extended KNN based

This approach is similar to the Naive KNN based approach but we introduce the external financial domain documents in this case. We consider all the documents and run Hearst Patterns [Hearst, 1992; Seitner *et al.*, 2016] on it. We get a database of automatically extracted hyponymy-hypernymy relations in financial domain form this. We make use of these extracted relations to infer relations for concepts during test time. We hypothesize that the input financial concept whose hypernymy is to be predicted is present in the automatically extracted database. However the exact term match would be crude way to do so. So we use word embedding based similarity to get a perfect match. even if there is an exact match, word embedding based similarity would give the highest score in that case.

For a given test financial concept, we take its average pre-trained embedding. We also consider average pre-trained embedding of all the hyponymys present in automatically extracted database. We compare both these embeddings using cosine similarity. We find the most representative hyponymy from automatically extracted database. This can be thought of as KNN with $k = 1$. We then take this representative hyponymy and compare its hypernymy with our set of labels. This is again done by taking average pre-trained embeddings and taking cosine similarity. This gives us the most similar label from the set.

2.5 Graph Based

We again make use of the automatically extracted hyponymy-hypernymy relations from external financial domain text. This database can contain different type of entities which may not be the exact match with concepts of our interest. There can be several hops in relations before we finally reach the parent hypernymy. For example "Equity Linked Bond" is a "Variable Coupon Bond" which in turn is a "Bond". Therefore we have to traverse the relations completely in order to get the broader picture. For this effect we turn to a graph based approach. We build a graph with entities as nodes and relations as edges. These entities come form the automatically extracted hypernymy-hyponymy database done using Hearst Patterns [Hearst, 1992; Seitner *et al.*, 2016]. For each relations we add an undirected edge.

We leverage the connections in the graph to predict the hypernymy of the financial concept. For the input financial concept we find a representative node using cosine similarity among the average pre-trained embeddings. Once we get the representative node we consider the connected component of the graph containing that representative node. The intuition is that the hypernymy label should be present in one of the nodes in this connected component. This is because of the whole taxonomic structure and relations among entities. So we only consider the connected component containing the representative node and find the hypernymy label node in it.

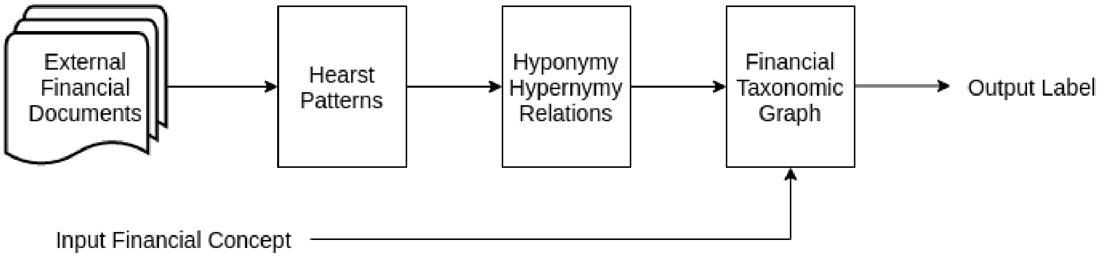
External Financial Documents
Hearst Patterns
Hyponymy Hypernymy Relations
Financial Taxonomic Graph
Output Label
Input Financial Concept

Figure 2: Overview of Graph Based Approach

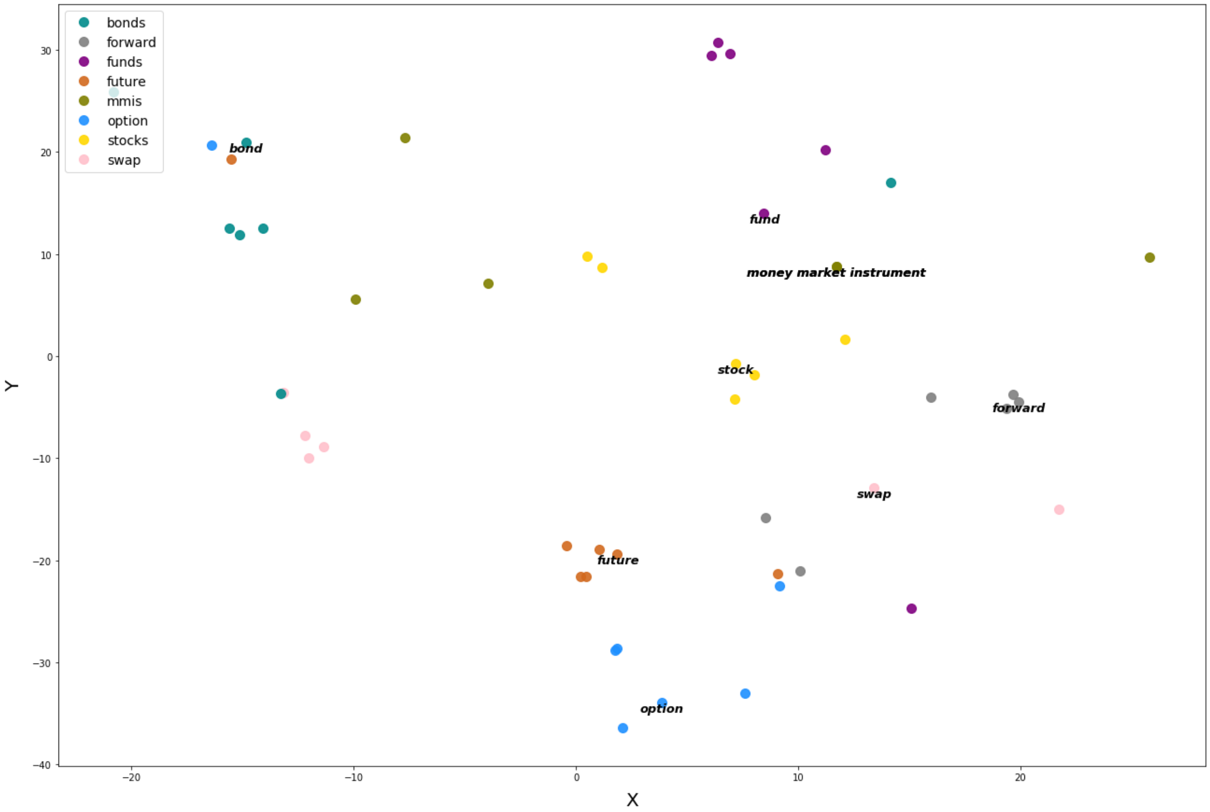

Figure 3: Illustration of financial concept USE embeddings using t-SNE

For each possible label we compute similarity scores again using average pre-trained embedding. We consider the maximum similarity score that we get when comparing each node with the label and assign it to that particular label. This way we will have scores for each of the label. Label with the maximum score is given as the prediction. Figure 2 gives the overview of the overall graph based approach.

3 Experiment

For pre-trained embeddings, we use 100 dimensional GloVe word embeddings and 512 dimensional USE sentence embeddings. We trained all deep learing architechure using Adam optimizer [Kingma and Ba, 2014] with 0.001 learning rate. For Naive KNN based we use the $K = 10$ as it gave best results for this method. We use hearstPatterns python library for the implementation of hearst patterns. It was used the extended mode to mine additional patterns. NetworkX python library was used for implementation of graph based algorithms.

4 Results and Discussion

Table 1 summarizes the results for various methods described above. USE embedding based cosine similarity gave the best results in both metrics - mean rank and accuracy. Deep learning based architectures also gave similar good scores while graph based methods didn't perform well.

Method	Mean Rank	Accuracy
GloVe	1.84	0.63
GloVe (fine-tuned)	1.79	0.67
GloVe (FIBO Description)	2.07	0.43
USE	**1.43**	**0.79**
USE (FIBO Description)	2.08	0.46
CNN	1.44	0.77
LSTM	1.44	0.78
RoBERTa	1.45	0.78
Naive KNN Based	1.75	0.61
Extended KNN based	3.80	0.08
Graph Based	2.68	0.19

Table 1: Results Table. Glove and USE embedding methods are Cosine Similarity based.

5 Conclusion

This paper mainly discusses how we tackle the FinSim 2020 shared task. The task is to automatically map financial concepts with its hypernymy. For this purpose we we explore the different ways in which we can learn the semantics in financial document for automatically predicting the hypernymy relations of financial concepts. We explore how pre-trained word and sentence embeddings can be used for this task. We experiment with both traditional and current deep learning architectures. We further explore how external financial documents can be useful. Our best method accomplishes good results for the task and puts us in one of the top positions among other participants of the task.

References

[Bordea *et al.*, 2016] Georgeta Bordea, Els Lefever, and Paul Buitelaar. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2016.

[Camacho-Collados *et al.*, 2018] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. Semeval-2018 task 9: Hypernym discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018); 2018 Jun 5-6; New Orleans, LA. Stroudsburg (PA): ACL; 2018. p. 712–24.* ACL (Association for Computational Linguistics), 2018.

[Cer *et al.*, 2018] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[Hearst, 1992] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[Seitner *et al.*, 2016] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 360–367, 2016.