EasyChair Preprint

№ 9878

Learning to Rank Hypernyms of Financial Terms using Semantic Textual Similarity

Sohom Ghosh, Ankush Chopra and Sudip Kumar Naskar

August 17, 2023

Learning to Rank Hypernyms of Financial Terms using Semantic Textual Similarity

Sohom Ghosh[*,1], Ankush Chopra[†,2] and Sudip Kumar Naskar[3]

[1]Fidelity Investments, Bengaluru, Karnataka, India, ORCiD: 0000-0002-4113-0958.
[2]Tredence Analytics, Bengaluru, Karnataka, India, ORCiD: 0000-0002-9970-8038.
[1,3]Jadavpur University, Kolkata, West Bengal, India, ORCiD: 0000-0003-1588-4665.

Contributing authors: sohom1ghosh@gmail.com; ankush01729@gmail.com; sudip.naskar@gmail.com;

Abstract

Over the years, with the advancement of digitalization, investors have started embracing the online mode of performing financial activities. Most investors prefer to read contents over the internet before making decisions. The financial services industry has terms and concepts that are complex and difficult to understand. In order to fully comprehend these contents, one needs to have a thorough understanding of these terms. Getting a basic idea about a term becomes easy when it is explained with the help of the broad category to which it belongs. This broad category is referred to as *hypernym*. In this paper, we propose a system capable of extracting and ranking hypernyms for a given financial term. The system has been trained with financial text corpora obtained from various sources. Embeddings of financial terms have been extracted using domain specific embeddings and fine-tuned using SentenceBERT [44]. A novel approach has been used to augment the training set with negative samples. Finally, we benchmark the system performance with that of the existing ones. We establish that it performs better than the existing ones and is also scalable.

Keywords: Hypernym Ranking, Text Similarity, Financial Texts, Natural Language Processing

.

[†]This work was done when Ankush was previously associated with Fidelity Investments, India

This paper is an extension of the solution [15] presented by our team LIPI at FinSim-3 [26] (FinNLP-2021 - workshop of IJCAI-2021)

*Corresponding Author

This pre-print has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this article is published in Springer Nature Computer Science, and is available online at https://doi.org/10.1007/s42979-023-02134-z

1 Introduction

Investors read online content (like financial reports of organizations, news) to make decisions. These contents often contain jargon unknown to the readers. The readability of these contents can be improved significantly by presenting readers with hypernyms (i.e. board categories) corresponding to any jargon. A jargon being a subset holds an "IS A" relationship with its hypernym. For example, "alternative debenture" (unknown financial term/jargon) is a kind of "bond" (hypernym). The same holds true for terms like "Bearer Bonds", "Callable Bonds" and "CoCo Bonds". This is shown in Figure 1. The Natural Language Processing (NLP) community has been working on methods to automatically discover hypernyms for more than a decade. Recently with the advent of shared tasks like FinSim [35] extracting hypernyms specific to the financial domain has caught the attention of this community. Inspired by the advances and contributions made by the participants in FinSim-1 [35] and FinSim-2 [36], we participated in the third edition of FinSim [26]. It comprised of matching financial terms to their hypernyms. Compared to the previous two editions, the third edition consisted of larger and more diverse topics related to finance. In this paper, we present an extension of the solutions our team LIPI developed while participating in FinSim-3 as well as the enhancements we carried out later.

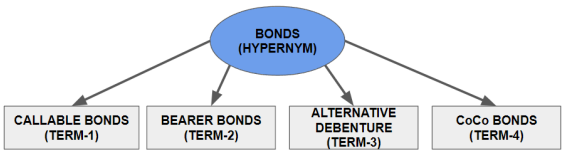

Fig. 1: Terms to Hypernym relation.

The research questions we try to answer in this study are as follows.

- **RQ1:** How have the datasets and solution architectures of the FinSim challenges evolved over the years?
- **RQ2:** How to develop a system for ranking a set of hypernyms for a given financial term?
- **RQ3:** Does using domain specific embeddings improve model performance?
- **RQ4:** What is the impact of augmenting/adding data from other sources?

Our contributions in the work contained in this article are as follows.

- We review and summarize various approaches used by participants of all three editions of FinSim [26, 35, 36]. We further collate the performances of such approaches in Table 2.
- We explore various external financial data sources to supplement the training set.
- We propose a novel way of augmenting the training set for incorporating hierarchies that are present in the set of hypernyms.
- We develop a system capable of ranking a set of hypernyms for a given financial term.

The data set used in this paper can be obtained from here[1]. The metadata is presented in the paper [26]. Our code base is available here [2].

This paper is organized as follows. Section 1 introduces readers to our motivation. Section 2 briefly narrates the previous works on this task. We formally define the problem statement in Section 3 and discuss the dataset used for this work in Section 4. Next, we describe our methodology, experiments and results in Section 5, 6 and 7 respectively. Section 8 concludes the paper and section 9 provides avenues for future work.

2 Research Landscape

In this section, we discuss the previous works in three phases. Firstly, we explore how the problem of hypernym identification have been solved in the field of computational linguistics in general. Following this, we elaborate its applications specific to the Financial Domain. Finally, we state how our work differs from the existing work in the literature.

2.1 Hypernym Identification in NLP Literature

The task of Hypernym detection started gaining the interest of the NLP community in early 1990.

[1] https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp2021/shared-task-finsim

[2] https://github.com/sohomghosh/FinSim_Financial_Hypernym_detection

During this time Hearst et al. [23] did the pioneering work of automatically extracting hypernyms using lexico-syntactic patterns like "such as" followed and preceded by Noun Phrase and so on. Another pattern-based approach had been applied by Snow et al. [48]. They narrated how they extracted "dependency paths"from parse trees of sentences containing hypernyms and hyponyms using WordNet [38]. They additionally used coordinate terms i.e. terms having at least one common parent to enhance the process of hypernym identification. Sang [52] assumed that the web contained much more data than any of the text corpora and developed a simple pattern-based method to extract hypernyms from the web. Furthermore, Sang et al.[53] compared two major approaches of hypernym extraction which are based on lexical (dictionary-based) and dependency patterns. Ritter et al. [45] described how they used lexical based patterns and Hidden Markov Models to identify hypernyms of noun phrases.

Caraballo [12] presented an automatic method of building a hierarchy of nouns and their hypernyms using WordNet [38]. Bottom-up clustering had been used to create the hierarchy and hypernyms had been assigned after creating a binary tree. Shinzato et al. [47] proposed a novel method of extracting hypernyms from web pages using structures of the HTML pages and other statistical features. Navigli et al. [39] introduced a novel concept of Word-Class Lattices which were learned from definitions present in Wikipedia. They further released a Java-based tool [18] to extract hypernyms of a term and its' definitions.

Recently, the use of Deep Learning Models in Computational Linguistics has gathered the interest of the NLP community. Tan et al. [50] used bi-directional Recurrent Neural Networks to extract hypernyms from definitions using Parts of Speech of constituent words. They validated this model's performance on Wikipedia as well as Stack-Overflow datasets. Liang et al. [31] studied if the property of transitivity holds in lexical taxonomies which were built automatically. They developed a supervised approach to do so. Furthermore, they used transitivity to extract new hypernym-hyponym relations.

2.2 SemEval Shared Tasks on Hypernym Detection

Problems relating to hypernym detection were provided in several editions of SemEval [5, 9–11].

SemEval-2015 Task 17: "Taxonomy Extraction Evaluation (TExEval)" [9] dealt with extraction of hypernym-hyponym relations from texts and taxonomy construction for four different domains namely: chemicals, equipment, foods and science. Grefenstette [21] developed the best performing model using simple structure-based features like whether a term is present in a sentence and document, term and document frequencies and presence of sub-sequences.

SemEval-2016 Task 13: "Taxonomy Extraction Evaluation (TExEval-2)" [10] was the multilingual edition of TExEval [9]. It comprised corpora from several domains like environment, food and science. Different languages included English, Dutch, Italian and French. Team Taxi [41] won both the shared tasks. They used Hearest pattern and sub-string based features.

SemEval 2017 Task 10: "ScienceIE - Extracting Keyphrases and Relations from Scientific Publications" [5] dealt with extraction of important phrases (like Process, Task and Material) and relations (like hypernyms / synonyms). It was restricted to the scientific domain. Team MIT [30] achieved the first rank by creating a system using a convolutional neural network. This system used an embedding comprising relative positions, type of entity and parts of speech as input.

SemEval-2018 Task 9: "Hypernym Discovery" was introduced [11] in the year 2018. This shared task was about extracting hypernyms from corpora in three languages (English, Spanish and Italian) and two domains within English (Medical and Music). The best performing model was presented by Team CRIM [8]. This model was an ensemble of word embedding based supervised approach with a pattern based unsupervised approach.

Dash et al. [16] introduced a new neural network-based architecture, Strict Partial Order Networks (SPON) to detect hypernyms. They benchmarked it using SemEval 2018 general and domain specific hypernym discovery tasks. Very recently Bai et al. [6] proposed the use of sequential recurrent mapping models to preserve the hierarchy between terms and their hypernyms.

They also performed an extensive evaluation on SemEval-2018 Task 9 datasets.

2.3 FinSim Shared Tasks - Hypernym Detection in Financial Texts

As mentioned earlier, the third edition of FinSim challenge [26] is the most recent one. Details relating to all editions of FinSim is mentioned in Table 1. These shared tasks have been organized by Fortia Financial Solutions[3]. Teams IITK [27], PolyU-CBS [14] and MXX [29] won the first, second and third editions of FinSim respectively. We shall narrate more details relating to the dataset of FinSim-3 in the next section 4. Team MXX [29] used a LSTM [25] based approach over word2vec [37] embeddings to win the FinSim-3 challenge (Accuracy = 1.113, Mean Rank = 0.941). The evaluation metrics and the other aspects of the problem statement remained the same for all three editions. We organize the system descriptions of the participating teams and present them in Table 2. The winning entries have been highlighted in bold. Studying this table thoroughly, we observe that the Word2Vec approach remained the same for all of them. Only one of these teams MXX [29] augmented the given dataset with external data. Similarly, only one of the winning team PolyU-CBS used syntactic based features like Jaccard similarity. Logistic Regression emerged out to be the most preferred classifier. Moreover, it is interesting to note that every successive year performances of the submitted models improved significantly. Since only three teams ([43], [49] and [19]) used Knowledge Graphs, we conclude it is yet to become popular. Some of the BERT based models like FinBERT [3], Sentence BERT [44] and RoBERTa [32] were also explored by most participants.

In recent times, Loukas [34] released the EDGAR-CORPUS comprising annual reports of listed US organizations from the year 1993 to 2020. They created word2vec [37] embeddings based on this corpus and evaluated it on the FinSim-3 dataset. They achieved an accuracy of 0.879 and a mean average rank of 1.21 using stratified 10-fold cross-validation.

[3] https://www.fortia.fr/

2.4 Difference with Prior Works

Our work is novel in terms of the approach we used to create negative samples from the existing dataset using the hierarchy present within the hypernyms. Unlike most others, we did not train a classifier to solve the problem of detecting hypernyms. On the other hand, we detect hypernyms by performing semantic search over fine-tuned embeddings. This makes the approach generic and robust to adding more hypernyms to the existing set.

3 Problem Statement

In this section, we shall narrate the problem statement and discuss the evaluation metrics.

Given a set of n financial terms (t_1, t_2, t_3, ... t_n) and their corresponding hypernyms/labels (l_1, l_2, l_3, ... l_n) where $l_i \epsilon$ {Equity Index, Regulatory Agency, Credit Index, Central Securities Depository, Debt pricing and yields, Bonds, Swap, Stock Corporation, Option, Funds, Future, Credit Events, MMIs, Stocks, Parametric schedules, Forward, Securities restrictions}. Our task is to develop a system capable of ranking all these hypernyms in order of decreasing semantic similarity for any unknown financial term.

The evaluation metrics used here are as follows:
$Accuracy = \frac{1}{n} * \sum_{i=1}^{n} I(y_i = \hat{y}_i[1])$,
$MeanRank = \frac{1}{n} * \sum_{i=1}^{n} (\hat{y}_i.index(y_i))$,
where $\hat{y}_i$ is the ranked list (with the index starting from 1) of predicted labels corresponding to the expected label y_i. I is an identity matrix. Interestingly, the organizers considered only the first three elements of the ranked list for evaluation. If any label was not present within these three elements, it was assigned rank 4.

4 Dataset

In this section, we narrate the datasets we used to perform our experiments. In addition to the data, which was provided to us by the organizing team, we explored other external datasets as well. These include Financial Industry Business Ontology (FIBO)[4], DBpedia[4], Investopedia[5], etc.

[4] https://spec.edmcouncil.org/fibo/
[5] https://www.investopedia.com/

Table 1: Background. #Pps is number of Prospectus. #L, #T, Acc. and MR. denote number of Labels, Teams, Best Accuracy and Mean Rank respectively.

Year	Edition	Conference	#Pps	#Train	#Test	#L	#T	Acc.	MR.
2020	FinSim-1 [35]	IJCAI-PRICAI	156	100	99	8	6	0.858	1.21
2021	FinSim-2 [36]	ACM-WWW	203	614	211	10	7	0.906	1.189
2021	FinSim-3 [26]	IJCAI	211	1050	326	17	5	0.941	1.113

4.1 Data Description

The organizers provided us with 211 prospectuses of different companies in Portable Document Format (PDF). Furthermore, a tagged dataset comprising 1050 financial terms and their corresponding hypernyms/labels were also provided. Out of 1050 terms, 1040 were distinct. We refer to this as the training set. Three of these terms were ambiguous as they were assigned 2 different labels. Terms with lengths less than or equal to six constituted 91% of the training set. Few instances of such terms are: 'Floating Rate Note'", "Perpetual bond", etc. Number of distinct labels was 17. Their distribution is shown in Figure 2 and presented in Table 3. It is interesting to note that a hierarchy was present among these 17 labels as all of them belonged to FIBO. This hierarchy is presented in Figure 3. The root nodes and leaf nodes have been highlighted in yellow and grey respectively. The first child nodes have been marked in bold. Moreover, we received 326 unlabeled financial terms which constituted the test set. The hypernym "Swap" share same parents with "Option" whereas it does not have any relation with other hypernyms like "Future" or "Bonds".

4.2 Data Augmentation

Since 91% of the training set had financial terms having only six or fewer words, we explored various ways of augmenting the dataset. Similar approach was also followed by [42] and [46] while participating in FinSim-2 and FinSim-1 respectively. This was done in three phases. Let us understand each one of them.

4.2.1 Acronym Expansion

Several Financial Terms were present along with their acronyms. This led to inconsistency in the training set. Keswani et al. [27] also highlighted this issue. To deal with this, we executed spaCy's[6] inbuilt acronym detector on all the prospectuses. We manually investigated the outputs (i.e., a list of acronyms and their corresponding synonyms). We concluded that not all of outputs were usable. We developed the following heuristics to clean this list further. We dropped records having

- expansions with number of characters lesser than that of the acronyms
- expansions with parenthesis/bracket symbols i.e., "(" or ")"
- expansions with number of characters lesser than or equal to five
- acronym which was a valid English word including proper nouns like "bond", "England", "Germany" and so on.

The cleaned list comprised 635 acronyms and their expansions. We used this list to augment our training set by replacing acronyms with their full forms wherever possible.

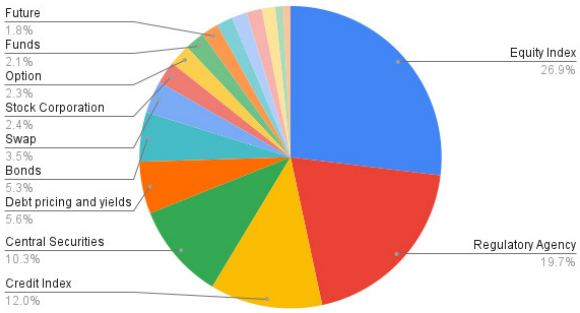

Fig. 2: Distribution of labels in original training set.

[6] https://spacy.io/

Table 2: Related Works - FinSim. USE: Universal Sentence Encoder, RF: Random Forest, LR: Logistic Regression, LSTM: Long Short Term Memory; NB: Naive Bayes, NN: Neural Networks, DA: Deep Attention, KG: Knowledge Graphs, SVM: Support Vector Machine, Inv: Investopedia, Ext: External.

				Approach of best performing model				
Task	**Team**	**Acc.**	**MR**	**Syntatic Features**	**Classifier**	**Embeddings**	**KG**	**Ext. Data**
FinSim-1	Anuj [46]	0.858	1.42	Character count, Word Count etc.	SVM			Inv
FinSim-1	ProsperaMnet [7]	0.777	1.34			Sparse embeddings		
FinSim-1	FINSIM20 [2]	0.787	1.43			USE [13]		
FinSim-1	**IIT-K** [27]	**0.858**	**1.21**		NB	Word2Vec, BERT		
FinSim-2	AIAI [51]	0.877	1.278		DA	Word2Vec [37]		
FinSim-2	FinMatcher [43]	0.811	1.415	Word overlap	NN		RDF2vec	WordNet, Wikidata, WebIsALOD
FinSim-2	GOAT [42]	0.896	1.193		LR	FinBERT [3]		Inv
FinSim-2	L3i-LBPAM [40]	0.858	1.325			SentenceBERT [44]		
FinSim-2	TCSWTIM2021 [20]	0.858	1.274	Sentence extraction		TF-IDF, BERT [17]		
FinSim-2	JSI [49]	0.811	1.316		RF	Word2vec	FIBO	
FinSim-2	**PolyU-CBS** [14]	**0.906**	**1.189**	Jaccard similarity, if hypernym in term	LR	Word2Vec, BERT		
FinSim-3	DICoE [33]	0.904	1.162	Levenshtein distance, Upper case to lower case characters ratio	LR	Word2vec		Inv, FIBO
FinSim-3	MiniTrue [19]	0.865	1.315			BERT, FinBERT	RotatE	
FinSim-3	Lipi [15] *(Our old model)*	0.917	1.156	Acronym expansion		SentenceFinBERT		DBPedia, Inv, FIBO
FinSim-3	Yseop [1]	0.917	1.141		LR	FastText, SentenceRoBERTa		FIBO
FinSim-3	**MXX** [29]	**0.941**	**1.113**		LSTM	Word2Vec		Inv,FIBO, NYSE, BIS

Table 3: Distribution of labels in the original training set.

Label	Count
Equity Index	280
Regulatory Agency	205
Credit Index	125
Central Securities Depository	107
Debt pricing and yields	58
Bonds	55
Swap	36
Stock Corporation	25
Option	24
Funds	22
Future	19
Credit Events	18
MMIs	17
Stocks	17
Parametric schedules	15
Forward	9
Securities restrictions	8
Total	**1040**

4.2.2 Augmenting definitions from DBpedia

DBpedia[7] provides search Application Programming Interfaces (API)[8] which helps in extracting structured information and relationships from Wikipedia[9]. Kilger [28] introduced The Linked Hypernyms Dataset which provided more specific details than DBpedia. We explored DBpedia extensively to obtain definitions of financial terms present in the training and test sets. These definitions added more context to the original terms. We present the results of invoking the search API for the term, "callable bond" in Figure 4. Inspecting some of these sample outputs manually, we concluded that we needed to match the given financial terms with the content of the "Label" tag present in the output payloads and extract the contents of the "Description" tag. To achieve this, we pre-processed the given financial terms and the contents of the "Label" tag obtained by calling the search API for each of the terms. The pre-processing steps included conversion to lower case, punctuation and repetitive white space replacement and singularization. Furthermore, we calculated the token overlap ratio between these cleaned terms and contents of the "Label" tag using these formulas:
$Ratio1 = length(s_1 \cap s_2)/length(s_1)$,
$Ratio2 = length(s_2)/length(s_1)$
where s_1 and s_2 represents sets of tokenized cleaned financial terms and tokenized cleaned contents of the "Label" tag respectively. After experimenting with several values, we empirically decided to use $Ratio1 = 1$ and $Ratio2 <= 1.25$. This enabled us to extract the descriptions of the matching terms from DBpedia.

4.2.3 Augmenting definitions from Investopedia and FIBO

While participating in FinSim-1, Saini [46] used definitions of financial terms from Investopedia[10]. Inspired by his approach, we crawled all these definitions from Investopedia. A total of 6,261 definitions were obtained. Moreover, we obtained a glossary of 11,827 financial terms and their explanations from FIBO. We cleaned these using the approach mentioned previously.

These data augmentation steps increased the size of the training set to 1836 records and the size of the test set to 607 records. For the financial term "callable bond" we present the result of data augmentation in Table 4. Table 5 presents the number of matches we get from different sources of data like DBPedia, Investopedia and so on.

4.2.4 Adding data from various external sources

Inspired by [29], we extracted 31,748 financial terms from various other websites such as

- Bank of International Settlements[11] (for label "Regulatory Agency")
- ETF Database[12] (for label "Equity Index")
- Wikipedia[13] & Wiley[14] (for label "Credit Index")
- Kaggle[15] (for label "Funds")

[7] https://www.dbpedia.org/
[8] https://lookup.dbpedia.org/api/search
[9] https://en.wikipedia.org/

[10] https://www.investopedia.com/financial-term-dictionary-4769738
[11] https://www.bis.org/regauth.htm
[12] https://etfdb.com/indexes/equity/
[13] https://en.m.wikipedia.org/wiki/Credit_default_swap_index
[14] https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119208631.app1
[15] https://www.kaggle.com/stefanoleone992/mutual-funds-and-etfs/version/3?select=MutualFunds.csv

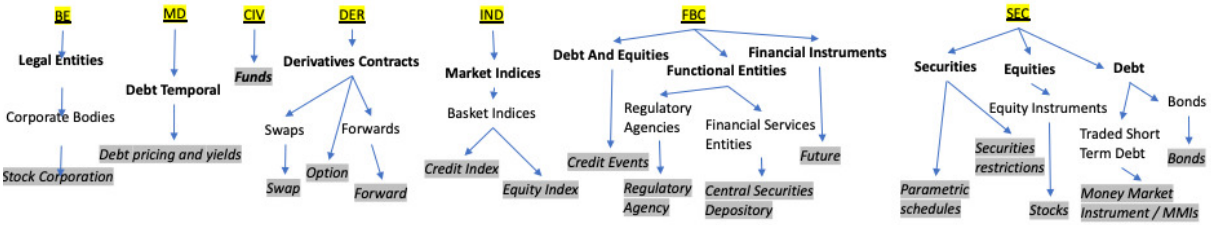

Fig. 3: Hierarchy of labels as obtained from FIBO. Root nodes have been underlined and highlighted in yellow. First child nodes have been marked in bold. Leaf nodes have been italicised and highlighted in grey color. BE = Business Entities, MD = Market Data, CIV = Collective Investment Vehicle, DER = Derivatives, IND = Indices and Indicators, FBC = Financial Business and Commerce, SEC = Securities


```
<Result>
  <Label>Callable bond</Label>
  <URI>http://dbpedia.org/resource/Callable_bond</URI>
  <Description>A callable bond (also called redeemable bond) is a type of bond (debt security) that allows the issuer of the bond to retain the privilege of redeeming the bond at some point before the bond reaches its date of maturity. In other words, on the call date(s), the issuer has the right, but not the obligation, to buy back the bonds from the bond holders at a defined call price. Technically speaking, the bonds are not really bought and held by the issuer but are instead cancelled immediately.</Description>
  <Classes/>
  <Categories>
    <Category>
      <URI>http://dbpedia.org/resource/Category:Bonds_(finance)</URI>
    </Category>
    <Category>
      <URI>http://dbpedia.org/resource/Category:Embedded_options</URI>
    </Category>
    <Category>
      <URI>http://dbpedia.org/resource/Category:Options_(finance)</URI>
    </Category>
  </Categories>
  <Refcount>4</Refcount>
</Result>
```


Source: https://lookup.dbpedia.org/api/search?query=callable%20bond

Fig. 4: Result obtained by calling DBPedia Search API for the term "callable bond".

- ADVFN[16] & datahub[17] (for label "Stock Corporation")
- National Securities Depository Limited[18] and European Central Securities Depositories Association[19] (for "Central Securities")

We added these terms to our training set for some of the experiments we performed. Later, we discarded them as it did not result in any improvement in the model performance. This is probably because most of these terms are proper nouns as they represent names of funds, organizations and so on.

4.3 Development, Validation and Test splits

As mentioned previously, we were provided with 1040 distinct manually tagged financial terms for training our model and 326 un-tagged instances for testing. We split the set of 1040 terms into two buckets: a development set having 831 terms (80%) and a validation set having 209 terms (20%). We did the same for the augmented set having 1836 financial terms out of which 1785 were distinct. This resulted in a set of 1440 distinct terms for training & validation and a set of 345 distinct terms for testing. The final output i.e., predicted ranks of the given 17 labels on the test set was to be submitted for the initial set of 326 un-tagged instances. Thus, for the augmented test set we calculated the mean cosine similarity with each of the labels for multiple occurrences of a term. We ranked the labels based on these similarities.

The distribution of labels before ("original") and after data augmentation ("extended") is shown in Table 6.

5 Methodology

Our best performing model is an ensemble of two models. Each of these models has been developed in three steps.

1. negative sample creation (reference: Algorithm 1)
2. using sentence transformers to fine-tune embeddings having 768 dimensions
3. calculating cosine similarities between terms and hypernyms.

[16] http://www.advfn.com/
[17] https://datahub.io/core/nyse-other-listings
[18] https://nsdl.co.in/related/wrld.php
[19] https://ecsda.eu/members-2/list-of-members

Table 4: Result obtained by data augmentation for the term "callable bond".

Expanded Term/Term Definition	**Label**	**Source**
Callable bond	Bonds	original, acronym expansion
Bond that includes a stipulation allowing the issuer the right to repurchase and retire the bond at the call price after the call protection period	Bonds	FIBO
A callable bond (also called redeemable bond) is a type of bond (debt security) that allows the issuer of the bond to retain the privilege of redeeming the bond at some point before the bond reaches its date of maturity.	Bonds	DBpedia

Table 5: Number of matches obtained from various data sources.

Data Source	**Count**
Original modelling data	1040
Acronym expansion	218
DBpedia	257
Investopedia	85
FIBO	236

This has been depicted in Figure 5. Steps 1 and 3 are common for both models. In the second step, we use FinBERT [3] embeddings for the first model and FinISH [1] embeddings for the second model.

STEP-1: In the first step, we create negative samples from the existing training set having sets of terms 'T', labels 'L', term definitions 'TT' and label definitions 'LL'. The definitions of labels and terms are obtained through data augmentation. For instances where we are not able to augment anything to a given financial term, we keep the term definition the same as the term. For each term 't' having definition 'td', its corresponding label 'l' and label definition 'ld', present in the training set we first assign a similarity score of 1.0 to the ('td', 'ld') pair. After that, we extract root node 'ln' and first child node 'lc' of 'l'. We then randomly select 10 labels and their corresponding definitions from 'L' such that none of the selected labels and their corresponding terms is the same as 'l' and 't'. For each such label 'la' and label definition 'lnd', we assign similarity scores corresponding to each of the ('td', 'lnd') pairs. This similarity score is assigned a value based on the following conditions
i) value = 2.0*k when the first child of 'la' i.e. 'lac' is the same as 'lc'
ii) value = 1.0*k when only the root node of 'la' i.e. 'lan' is same as 'ln' and its first child 'lac' is different from 'lc'
iii) value = 0.0*k when former two conditions are not met i.e. they have no ancestors in common
We present this formally in Algorithm 1. We empirically determine that keeping the value of parameter k as 0.4 gives the best result. This resulted in 63,360 instances in total out of which 49,836 had a similarity score of 0.0. We subsampled the instances with similarity score of 0.0. The final distribution consists of 5,760 instances with a 1.0 similarity score, 5304 instances with 0.8, 2460 with 0.4 and 550 with a similarity score of 0.0. This step is common for both the models described above.

A machine learning based classification model learning model performs better when it is provided with more data from different classes. This motivated us to create negative samples. For example, as "Bonds" is the hypernym of "Alternate Debenture", we can safely assume that "Alternate Debunture" when paired terms having hypernyms other than "Bonds" will constitute negative instances,

STEP-2: In the second step, for the first model we fine-tune FinBERT [3] embeddings using sentence transformer [44] architecture. For the second model, we further fine-tune the FinISH

Table 6: Label distribution for the development and validation set before and after data augmentation.

	Original		Extended	
label	**# dev**	**# val**	**# dev**	**# val**
Equity Index	225	57	373	84
Regulatory Agency	159	46	260	78
Credit Index	103	21	123	27
Central Securities Depository	83	24	106	28
Bonds	49	6	110	14
Debt pricing and yields	41	17	84	34
Swap	31	5	57	9
Option	21	3	35	4
Stock Corporation	18	6	54	15
Funds	17	5	36	10
Future	16	3	29	7
Credit Events	15	3	35	6
Parametric schedules	14	1	45	3
MMIs	14	3	29	9
Stocks	12	5	23	11
Securities restrictions	7	1	28	3
Forward	6	3	13	3
TOTAL	**831**	**209**	**1440**	**345**

embeddings released by Yseop Labs[1]. They created this embedding by fine-tuning RoBERTa[32] on the FIBO corpus. Our objective was to minimize the multiple negative ranking loss and online contrastive loss. Multiple negative ranking loss [24] is applied only on samples which are similar to each other. This makes the embedding suitable for retrieval tasks. Online contrastive loss selects the hard cases in a batch based on the distance of separation and computes the loss only for these specific hard cases only. It tends to keep similar texts near to each other and pushes dissimilar texts away from each other in the vector space. We kept the margin parameter at 0.5. A batch size of 20, when executed for 25 epochs, gave the best result for the first model. For the second model, a batch of 30 when executed for 45 epochs gave the best result. The sample code is available here.[20].

STEP-3: In the third step, we convert definitions of all the 17 labels/hypernyms and terms present in the validation and test set into vectors. We use the fine-tuned embeddings generated in the previous step for the same. We further calculate cosine similarity between the vectors of each of these terms with that of all the 17 hypernyms. Since we have had augmented the dataset, we need to roll up this data such that we have only one record for every term. We use the mean of cosine similarities to achieve this. We do the same for the other model as well. This results in two cosine similarities for each of the terms one obtained from the first model while the other from the second.

To ensemble, we again take the mean of the two cosine similarities we calculated for each of the terms across all the hypernyms. Finally, we rank the hypernyms in terms of decreasing order of the mean cosine similarity.

6 Experimentation

In this section, we shall narrate various experiments we performed systematically to arrive at final model described in the previous section. We started by evaluating the baseline models provided to us.

[20] https://www.sbert.net/examples/training/quora_duplicate_questions/README.html#multi-task-learning (accessed on October 2021)

Algorithm 1 Algorithm to generate negative samples from existing training set

Require: $T > 0$ and $L > 0$ ▷ T is the augmented set of financial terms and L consists of corresponding labels i.e., hypernyms. $TT > 0$ and $LL > 0$ are the set of definitions of terms and labels respectively obtained after performing data augmentation

Require: Function FR(n) and Function FC(n) ▷ Function FR and FC returns the root node and first child node corresponding to node n respectively where n is one of the 17 labels i.e., leaf nodes/hypernyms

Ensure: $length(T) = length(TT) = length(L) = length(LL)$

$NT \leftarrow \{\}$ ▷ NT is the new set of definitions of financial terms to be created by appending negative samples

$NL \leftarrow \{\}$ ▷ NL is the new set of definitions of labels corresponding to terms in NT

$NS \leftarrow \{\}$ ▷ NS is the set of assigned similarity scores between the newly selected definitions of terms and labels in NT & NL respectively

$k \leftarrow 0.0$ ▷ 'k' is a hyper-parameter. Keeping k = 0.0 gives the best result

for each term t ∈ T, term definition td ∈ TT, corresponding label l ∈ L and label definition ld ∈ LL **do**

  $NT \leftarrow NT \cup \{td\}$

  $NL \leftarrow NL \cup \{ld\}$

  $NS \leftarrow NS \cup \{1.0\}$ ▷ Assign a similarity score of 1.0 as the term and the label definition belong to the original set

  $ln \leftarrow FR(l)$ ▷ Extract root node of label 'l'

  $lc \leftarrow FC(l)$ ▷ Extract first child node of label 'l'

  R, RR ε_r L, LL where length(R)=10, length(RR)=10 ▷ Randomly select 10 labels from 'L 'and corresponding label definitions from 'LL' ensuring none of the labels are 'l' and none of their corresponding terms is 't'. This is done for creating the negative set

  for each label la ∈ R and corresponding definition lnd ∈ RR **do**

    $NT \leftarrow NT \cup \{td\}$

    $NL \leftarrow NL \cup \{lnd\}$

    $lan \leftarrow FR(la)$ ▷ Extract root node of label 'la'

    $lac \leftarrow FC(la)$ ▷ Extract first child node of label 'la'

    if $lac = lc$ **then** ▷ Check if first child nodes are the same. This implies root nodes are also the same.

      $NS \leftarrow NS \cup \{2 * k\}$

    else if $lan = ln$ **then** ▷ Check if root child nodes are same when first child nodes are different

      $NS \leftarrow NS \cup \{1 * k\}$

    else ▷ When first child nodes and root nodes are different

      $NS \leftarrow NS \cup \{0 * k\}$

    end if

  end for

end for

return NT, NL, NS

6.1 Baselines

Let's understand the baseline solutions provided by the organizers. Kang et al. [26] trained a custom word2vec [37] model having 300 dimensions on text corpus extracted from the prospectus.

Baseline-1: In the first system, they calculate distances between terms and hypernyms based on the custom word2vec embeddings. They rank the hypernyms on the increasing order of distance.

Baseline-2: The second system comprises a logistic regression-based classifier trained using custom word2vec embeddings of the financial terms as independent variables and hypernyms as the dependent variables.

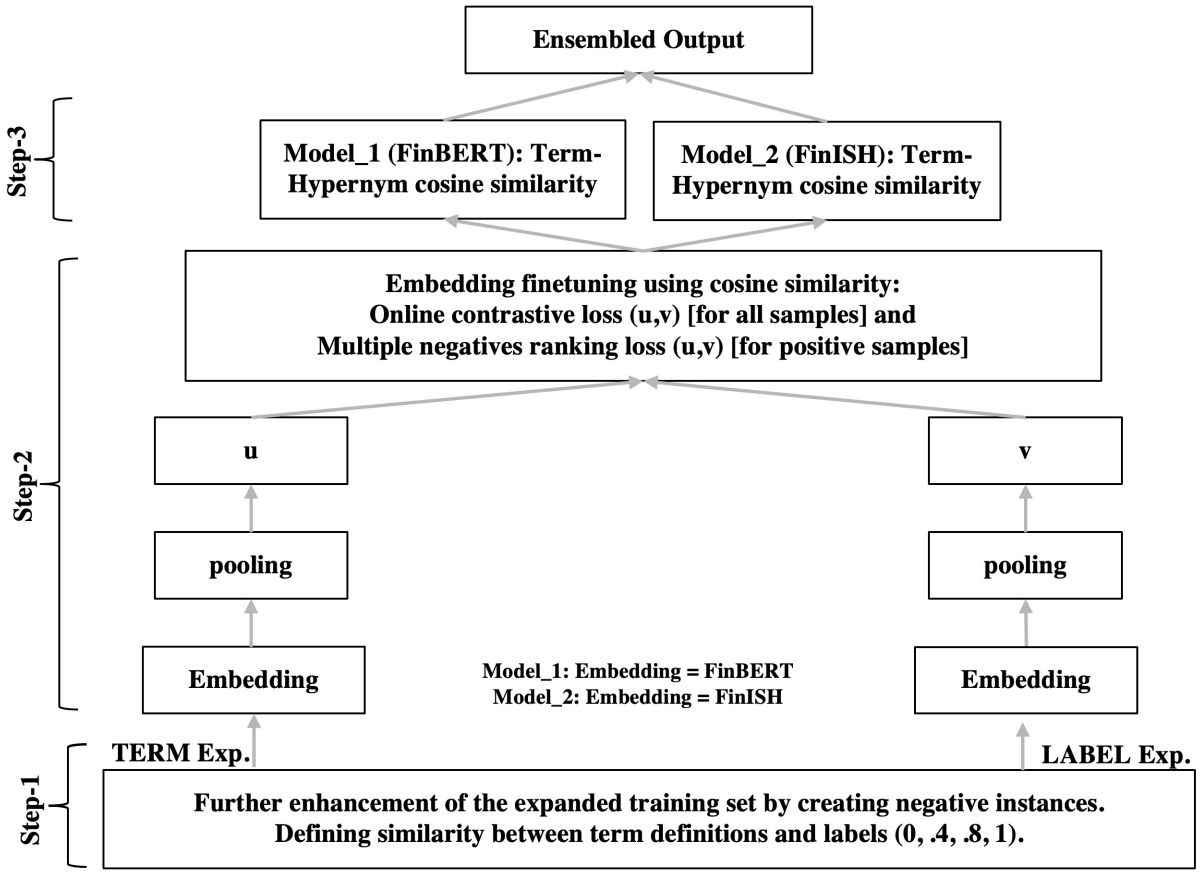

Fig. 5: Methodology

6.2 Experiments

At first, we removed the duplicate observations that we observed in the original dataset. We reserved 20% of the data for the unbiased validation set and the remaining 80% was used for training the models. We identified sources like DBpedia, FIBO and Investopedia which contain the definitions of many terms present in the input set. We also extracted the acronym definitions from the prospectus corpus shared by the organizers. All these sources helped us to augment the training data. The augmented data consisted of the original records along with the records where input terms were replaced with definitions and expansions. The number of instances in the original and the augmented training set was 832 and 1470. Similarly, the number of instances in the original and the augmented validation set was 208 and 366. This indicates we were not able to get a definition or expansion for each of the terms.

We began the experimentation by creating Term Frequency Inverse Document Frequency (TF-IDF) matrix, Topic Models and creating a machine learning based classifier over it. Since the performance was not appealing, we fine-tuned one of the state of the art pre-trained models known as BERT [17]. We used sub-word tokenization and followed the standard classification architecture to fine-tune the pre-trained models. We took the representation from [CLS] token and passed it to the feedforward layers. The last layer of the network had 17 nodes with SoftMax activation. These 17 nodes provided the prediction for 17 labels mentioned previously. We did not freeze the base model while training. This enabled the fine-tuning of the base model for the task at hand, resulting in better performance. During the training, the error was propagated back through the transformer network. Looking at the distribution of the tokenized output length, we decided to keep the maximum

input sequence length as 32. We ran extensive hyperparameter tuning and identified that a combination of Adam optimizer with a learning rate of 0.00002 and 64 batch size gave us the best result. We trained the model for 40 epochs with an early stopping criteria based on the performance on validation set. It performed the best after the 18^{th} epoch. We ordered the hypernyms in decreasing order of predicted probabilities. This performance was much better than that of the baselines.

We further tried the same BERT model in the augmented dataset which included the definitions from various sources mentioned previously. These definitions were well-structured sentences and they comprised longer sequences of input terms. We repeated the experiments described previously after increasing the input maximum input sequence length to 256. This input length was decided based on the distribution of the number of tokens that were present in the term definitions after the augmentation step. We trained it till the 40 epoch and found out that its performance on the validation set was best at the 17^{th} epoch. We observed that this performance was significantly better than that of the models developed without data augmentation. This led us to conclude that the data augmentation steps we followed were useful. We also tried adding data from various other sources as mentioned in section 4.2.4. However, this did not yield any further improvement in the performance of the model. This is probably because most of these terms were proper nouns and organization like entities.

We subsequently tried out various other transformer-based models present in the Huggingface [54] model repository. This included RoBERTa [32], FinBERT [3], FinEAS [22] and so on. We observed that FinBERT when fine-tuned using the expanded data set further improved the performance. Subsequently, we trained a new model based on transformer architecture. Its objective was to predict two things together i) root node ii) hypernyms. This did not perform well. We also tried to fine-tune these models using the Masked Language Model based approach on the corpus of the prospectus. Due to resource constraints, we could not train it beyond a few epochs. Its performance was not promising as well.

After extensively studying the failed cases and observing the hierarchy of the labels we decided to try out a novel framework to generate negative instances and fine-tune it, using the sentence transformer [44] architecture. This has been elaborated in detail in section 5. For creating the negative set mentioned in Algo: 1, we experimented with different sampling strategies and with various values of 'k'. The performance of the model improved when we used the sentence transformer architecture with FinBERT at the back end. It improved further on changing the base embedding from FinBERT to FinISH. FinISH was developed and resealed by Yseop Labs[21] while participating in FinSim-3 [1]. We ran it for 45 epochs with a batch size of 30. It took around 1 hour 43 minutes to train.

Finally, we tried to ensemble the best performing models. We observed that an ensemble of the last two models which were trained using sentence transformers architecture with negative samples resulted in the best performance on the validation set. All the hyper-parameters were selected empirically by tracking the model performance on the validation set.

We performed the experiments on Google Colab[22] (free tier) and on a Nvidia DGX GPU cluster. The cluster consists of 32 Nvidia Tesla V100 GPUs, over 160,000 CUDA cores and over 20,000 Tensor Cores. We used Python (3.7) for all the computations. The main libraries used here consists of PyTorch[23], SentenceTransformers[24], pandas[25], NumPy[26] and scikit-learn[27].

7 Results and Discussions

In this section, we shall discuss the results presented in Table 7. We restrict our evaluation to just one dataset due to non-availability of any other dataset suitable for financial hypernym detection.Models with serial numbers (SLN) 1 to 15 were developed during the FinSim-3 challenge while those with SLN 16 to 20 were developed later. After the event, the organizers declared the results for each submission of the participating teams. The number of submissions was restricted

[21] https://yseop.com/
[22] https://research.google.com/colaboratory/
[23] https://pytorch.org/
[24] https://www.sbert.net/
[25] https://pandas.pydata.org/
[26] https://numpy.org/
[27] https://scikit-learn.org/stable/

to 3. Thus, we present test set results for three of our models (SLN: 5, 6, 7). On comparing this with the test set results of other participants (SLN: 8 to 15), we observe that our old model SFinBERT_neg (SLN: 7) [15] ranked third and was marginally behind the one which was ranked second (SLN: 15) [1]. This model was developed by fine-tuning FinBERT [3] with negative samples using sentence transformer architecture. We tried reaching the organisers to evaluate our new model (SLN: 20) on the test set as well. However, the test set has not yet been released publicly. Thus, we present our results on the holdout validation set.

It is interesting to observe that on using transformer-based pre-trained BERT embeddings (SLN: 3, 4), the model performs better than the baselines (SLN: 1, 2). This proves the effectiveness of transformer-based embeddings like BERT [17] over traditional embeddings like word2vec [37]. It happened probably because transformer-based embeddings having been pre-trained on large datasets can capture more complexities within the language. Comparing the performance of models (having SLN: 3 and 5) with those (having SLN: 4 and 6) we conclude that external data augmentation has resulted in a performance gain. We also notice that financial domain specific embedding FinBERT [3] (SLN: 5, 6) resulted in improvement of the model performance when compared to generic embedding like BERT [17] (SLN: 3, 4). Furthermore, it is quite interesting to note that fine-tuning FinBERT [3] using a classifier layer to top (SLN: 5 and 6) to predict hypernym did not perform as good as fine-tuning a FinBERT model using sentence transformer where negative samples were also included (reference: SFinBERT_neg with SLN: 7). This is because several hypernyms were inter-dependent as shown in Figure 3.

Models with SLN 8 to 15 have been developed by other participating teams. Since their models were not open sourced, we are not able to present the performance of their models on our hold-out validation set. For the team MXX (SLN: 13), we quote the performance on their validation set as presented in the paper [29].We mentioned the approaches followed by other teams in Table 2. In the model SFinBERT_neg_th (SLN: 16) we changed 'k' (mentioned in section 5) from 0.4 to 0.2. The rest has been kept the same as the model SFinBERT (SLN: 7). Similarly, we tried changing the sampling strategy in the model SFinBERT_neg_ss (SLN: 17). Instead of sampling over the entire set 'L' (as mentioned in Algorithm 1), we tried considering all other hypernyms. Both methods did not improve the performance.

Moreover, in the model SFinBERT_neg (SLN: 7) we tried using FinISH embeddings instead of the FinBERT embeddings. We trained it for 45 epochs after increasing batch size to 30. This improved the model performance (Mean Rank: 1.072 and Accuracy: 0.952). We refer this model as SFinHyp_neg (SLN: 18). As mentioned in section 4.2.4, on adding more data to this model deteriorated the performance slightly. This is due to the fact this data is comprised mainly of proper nouns. We refer to it as Model SFinHyp_more_data (SLN: 19). Finally, ensembling models SFinHyp_neg (SLN: 18) with SFinBERT_neg (SLN: 7) resulted in the best performance (Mean Rank: 1.053 and Accuracy: 0.967). It performed even better than the old model we submmited at FinSim-3 (SLN: 7) and the existing state of the art model MXX (SLN: 13)on the held out validation set. We denote this ensemble model as Ensemble_7_18 (SLN: 20).

We further analyse the results for every label along with their root nodes. This is presented in Table 8. We notice that for all the labels having root node 'CIV', 'SEC' and for labels 'Forward', 'Option', 'Future', 'Credit Events' and 'Equity Index' the model performs the best. For the labels 'Stock Corporation', 'Swap' the proposed model performs the worst. For all other labels, the model performance is mediocre.

As a next step, we used Principal Component Analysis (PCA) to visualize the embeddings of the hypernyms generated using the method SFinHyp_neg (SLN: 18) in 2 dimensions. It is quite interesting to note that 'Option' and 'Future' despite having neither the root node nor the first child node in common are close to each other. This is because they are similar financial trading products. Thus, we can say the model captured the semantic aspect to some extent as well. We also observe that 'Regulatory Agency' and 'Central Securities Depository' which have the same root node 'FBC' are together. Similarly, hypernyms which do not have anything in common like 'Stock Corporation' and 'Debt pricing and yields' are separate from the rest. However, this is not the case for most other hypernyms. This is because

we are losing out on much information while projecting 768 dimensions of the embeddings to 2 dimensions. Our PCA model captures only 28.3% of the variance.

Ablation Study

To understand the significance of each component of our model (Ref: Figure 5) we do an ablation study. We present the results in Table 9. Analysing these results, we see that if we use readily available FinBERT embeddings [3] or fine-tuned RoBERTa embeddings [1] to simply rank the hypernyms based on cosine similarity with the financial terms and their definitions, then the performance deteriorates drastically. This explains the importance of the algorithm we developed to create negative sets. The final ensemble model performs better than the constituent models.

8 Conclusion

In this paper, we study the approaches followed by participants of all three editions of the FinSim challenge. Furthermore, we present a novel method of fine-tuning FinBERT [3] and FinISH [1] embeddings using hierarchies present in FIBO. This enabled us to rank a set of hypernyms for a given financial term. We conclude that pretrained transformer-based embeddings fine-tuned with domain specific data performed better in this scenario. We also observe that augmenting the existing data set with external data enhanced the model performance. However, adding more data like names of companies, mutual funds and stocks did not add any value.

While studying the stability of the model, we observe that during the training phase, we picked up random samples only in two places. During evaluation, we use two models to generate embeddings. These are further used to calculate cosine similarities between a given set of financial terms and hypernyms. The final ranking is done by taking mean of these two cosine similarities. Thus, the predictions generated from the ensemble model are stable.

Unlike the models developed by other participating teams ([29], [1] and so on), our model is not a classification model. Thus, we don't need to retrain it frequently if additional hypernyms are added. Moreover, the LSTM network which team MXX [29] trained cannot be parallelized and scaled. It won't be able to effectively deal with out-of-vocabulary words. It is easier to compute the mean of two cosine similarities than using two bi-directional LSTM networks to predict the hypernyms. This makes our model simple, scalable and easy to deploy when compared to that of the others.

9 Future Works

In future, we would like to gather more data for training and explore the use of Knowledge Graphs and Graph Neural Networks to improve these models. We also want to work on interpreting these models using various model explainability plots and participate in the upcoming challenges like FinSim-4[28]. Furthermore, an interesting direction for further research would be to create embeddings especially for financial terms and their definitions. Presently, we explored the hierarchies and relation trees present in FIBO. Although 'Future' and 'Options' are similar trading products, they are present in different trees. We would like to take this into account as well while creating our negative set. Using Neural Network based ranking loss may result in the better rank ordering of the hypernyms. Finally, we want to evaluate the statistical significance of predictions from these models over the baselines on a larger dataset.

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Declarations

Ethics approval

This research did not involve any human participants and/or animals. There was no need for informed consent.

Funding

Not Applicable.

[28] https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp-2022/shared-task-finsim4-esg

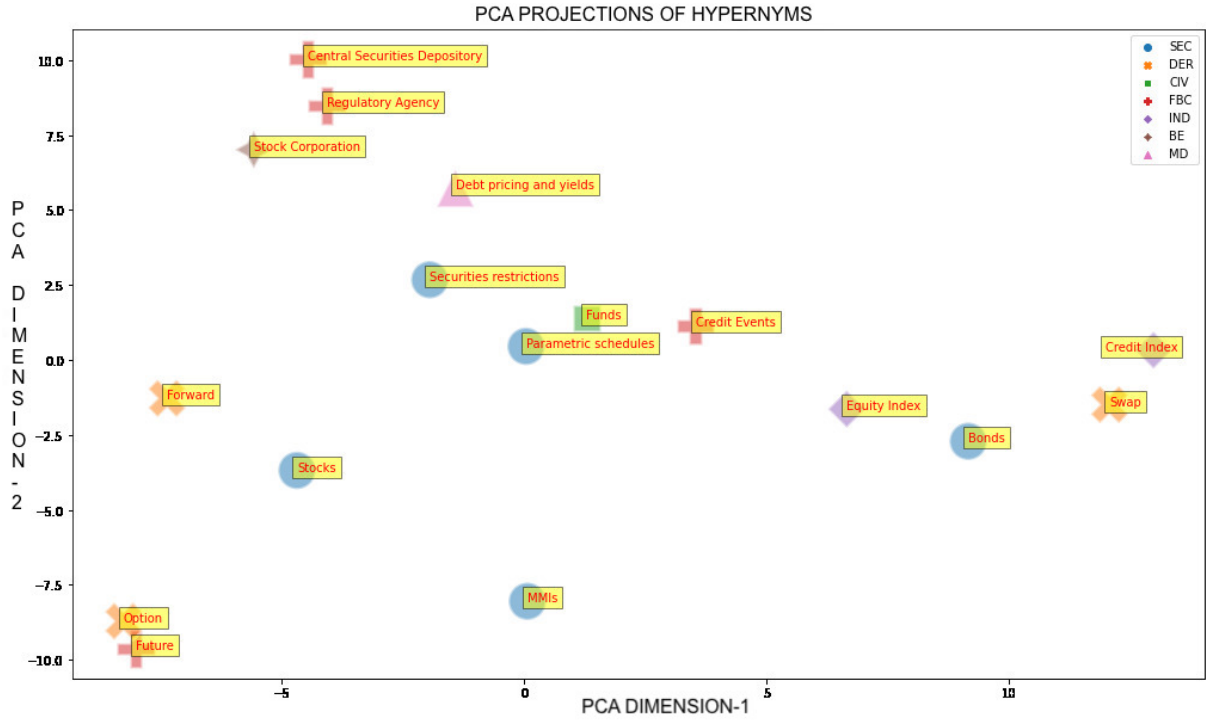

Fig. 6: PCA projection of embeddings of Hypernyms in 2 dimensions. Same shape denotes same root nodes.

Author contributions

Sohom Ghosh and Ankush Chopra conducted the experiments and prepared the manuscript. Sudip Kumar Naskar re-examined it. All authors reviewed the manuscript.

Availability of data and material

The data set used in this paper can be obtained from https://sites.google.com/nlg.csie.ntu.edu.tw/finnlp2021/shared-task-finsim. The metadata is presented in the paper [26].

Code availability

Our code base is available at https://github.com/sohomghosh/FinSim_Financial_Hypernym_detection.

Acknowledgements

We express our sincere gratitude to the organizers of FinSim-3 [26] for providing us with labelled data, evaluation scripts and starter codes.

References

[1] H. A. Akl, D. Mariko, and H. de Mazancourt. Yseop at FinSim-3 shared task 2021: Specializing financial domain learning with phrase representations. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 52–57, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.9.

[2] V. Anand, Y. Agrawal, A. Pol, and V. Varma. FINSIM20 at the FinSim task: Making sense of text in financial domain. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 104–107, Kyoto, Japan, 5 Jan. 2020. -. URL https://www.aclweb.org/anthology/2020.finnlp-1.17.

[3] D. Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A

Table 7: Results on validation and test set. Org. represents original and Ext. represents extended. Base refers to baseline. MR is Mean Rank.

			Validation Set		**Test Set**	
SLN.	**Model**	**Data Aug.**	**MR**	**Acc.**	**MR**	**Acc.**
1	Base-1	No	2.158	0.498	1.941	0.564
2	Base-2	No	1.201	0.876	1.750	0.669
3	BERT	No	1.177	0.899	-	-
4	BERT	Yes	1.153	0.928	-	-
5	FinBERT	No	1.117	0.928	1.257	0.886
6	FinBERT	Yes	1.110	0.942	1.220	0.895
7	SFinBERT_neg (Our old model) [15]	Yes	1.086	0.947	1.156	0.917
8	dicoe_1 [33]	No	-	-	1.180	0.889
9	dicoe_2 [33]	Yes	-	-	1.162	0.904
10	MiniTrue_2 [19]	No	-	-	1.315	0.865
11	MiniTrue_1 [19]	No	-	-	1.346	0.855
12	MiniTrue_3 [19]	No	-	-	1.337	0.825
13	mxx [29]	Yes	*1.06*	*0.96*	**1.113**	**0.941**
14	yseop_1 [1]	Yes	-	-	1.236	0.883
15	yseop_2 [1]	Yes	-	-	1.141	0.917
16	SFinBERT_neg_th	Yes	1.110	0.938	-	-
17	SFinBERT_neg_ss	Yes	1.105	0.933	-	-
18	SFinHyp_neg	Yes	1.072	0.952	-	-
19	SFinHyp_more_data	Yes	1.306	0.813	-	-
20	**Ensemble_7_18 (Our new Model)**	Yes	**1.053**	**0.967**	-	-

Table 8: Model performance for each labels CSD means Central Securities Depository.

Root	**Label**	**Mean Rank**	**Acc.**
BE	Stock Corporation	1.333	0.833
CIV	Funds	1.000	1.000
DER	Forward	1.000	1.000
DER	Option	1.000	1.000
DER	Swap	1.200	0.800
FBC	Future	1.000	1.000
FBC	Regulatory Agency	1.087	0.935
FBC	CSD	1.042	0.958
FBC	Credit Events	1.000	1.000
IND	Equity Index	1.000	1.000
IND	Credit Index	1.143	0.952
MD	Debt pricing and yields	1.059	0.941
SEC	Bonds	1.000	1.000
SEC	MMIs	1.000	1.000
SEC	Stocks	1.000	1.000
SEC	Parametric schedules	1.000	1.000
SEC	Securities restrictions	1.000	1.000

Table 9: Ablation Study on the validation set. cos. sim. means cosine similarity.

Model	**Mean Rank**	**Acc.**
Only FinBERT + cos. sim.	2.421	0.297
Only SFinHyp + cos. sim.	1.301	0.804
SFinBERT_neg	1.086	0.947
SFinHyp_neg	1.072	0.952
Ensemble (Our Model)	**1.053**	**0.967**

nucleus for a web of open data. In *The Semantic Web*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3540762973. doi: 10.1007/978-3-540-76298-0_52. URL https://doi.org/10.1007/978-3-540-76298-0_52.

[5] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada, Aug. 2017. Association for Computational

Linguistics. doi: 10.18653/v1/S17-2091. URL https://aclanthology.org/S17-2091.

[6] Y. Bai, R. Zhang, F. Kong, J. Chen, and Y. Mao. Hypernym discovery via a recurrent mapping model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2912–2921, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.257. URL https://aclanthology.org/2021.findings-acl.257.

[7] G. Berend, N. Kis-Szabó, and Z. Szántó. ProsperAMnet at the FinSim task: Detecting hypernyms of financial concepts via measuring the information stored in sparse word representations. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 98–103, Kyoto, Japan, 5 Jan. 2020. -. URL https://www.aclweb.org/anthology/2020.finnlp-1.16.

[8] G. Bernier-Colborne and C. Barrière. CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 725–731, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1116. URL https://aclanthology.org/S18-1116.

[9] G. Bordea, P. Buitelaar, S. Faralli, and R. Navigli. SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2151. URL https://aclanthology.org/S15-2151.

[10] G. Bordea, E. Lefever, and P. Buitelaar. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1168. URL https://aclanthology.org/S16-1168.

[11] J. Camacho-Collados, C. Delli Bovi, L. Espinosa-Anke, S. Oramas, T. Pasini, E. Santus, V. Shwartz, R. Navigli, and H. Saggion. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1115. URL https://aclanthology.org/S18-1115.

[12] S. A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034705. URL https://aclanthology.org/P99-1016.

[13] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder, 2018.

[14] E. Chersoni and C.-R. Huang. *PolyU-CBS at the FinSim-2 Task: Combining Distributional, String-Based and Transformers-Based Features for Hypernymy Detection in the Financial Domain*, page 316–319. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451387.

[15] A. Chopra and S. Ghosh. Term expansion and FinBERT fine-tuning for hypernym and synonym ranking of financial terms. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 46–51, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.8.

[16] S. Dash, M. F. M. Chowdhury, A. Gliozzo, N. Mihindukulasooriya, and N. R. Fauceglia. Hypernym detection using strict partial order networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications*

of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7626–7633, New York, USA, 2020. AAAI Press. URL https://aaai.org/ojs/index.php/AAAI/article/view/6263.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[18] S. Faralli and R. Navigli. A Java framework for multilingual definition and hypernym extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 103–108, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-4018.

[19] C. Feng and S. Wei. Exploiting network structures to improve semantic representation for the financial domain. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 58–62, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.10.

[20] T. Goel, V. Chauhan, I. Verma, T. Dasgupta, and L. Dey. *TCS WITM 2021 @FinSim-2: Transformer Based Models for Automatic Classification of Financial Terms*, page 311–315. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451386.

[21] G. Grefenstette. INRIASAC: Simple hypernym extraction methods. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 911–914, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2152. URL https://aclanthology.org/S15-2152.

[22] A. Gutiérrez-Fandiño, M. N. i Alonso, P. Kolm, and J. Armengol-Estapé. Fineas: Financial embedding analysis of sentiment, 2021.

[23] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, pages –, -, 1992. -. URL https://aclanthology.org/C92-2082.

[24] M. Henderson, R. Al-Rfou, B. Strope, Y. hsuan Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil. Efficient natural language response suggestion for smart reply, 2017.

[25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[26] J. Kang, I. E. Maarouf, S. Bellato, and M. Gan. FinSim-3: The 3rd shared task on learning semantic similarities for the financial domain. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 31–35, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.5.

[27] V. Keswani, S. Singh, and A. Modi. IITK at the FinSim task: Hypernym detection in financial domain via context-free and contextualized word embeddings. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 87–92, Kyoto, Japan, 5 Jan. 2020. -. URL https://www.aclweb.org/anthology/2020.finnlp-1.14.

[28] T. Kliegr. Linked hypernyms: Enriching dbpedia with targeted hypernym discovery. *Journal of Web Semantics*, 31: 59–69, 2015. ISSN 1570-8268. doi: https:

//doi.org/10.1016/j.websem.2014.11.001. URL https://www.sciencedirect.com/science/article/pii/S1570826814001048.

[29] N. Kroher, A. Pikrakis, S. White, and J. Lyske. MXX@FinSim3 - an LSTM–based approach with custom word embeddings for hypernym detection in financial texts. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 36–39, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.6.

[30] J. Y. Lee, F. Dernoncourt, and P. Szolovits. MIT at SemEval-2017 task 10: Relation extraction with convolutional neural networks. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 978–984, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2171. URL https://aclanthology.org/S17-2171.

[31] J. Liang, Y. Zhang, Y. Xiao, H. Wang, W. Wang, and P. Zhu. On the transitivity of hypernym-hyponym relations in data-driven lexical taxonomies. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1185–1191, San Francisco, California, USA, 2017. AAAI Press.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[33] L. Loukas, K. Bougiatiotis, M. Fergadiotis, D. Mavroeidis, and E. Zavitsanos. DICoE@FinSim-3: Financial hypernym detection using augmented terms and distance-based features. In *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing*, pages 40–45, Online, 19 Aug. 2021. -. URL https://aclanthology.org/2021.finnlp-1.7.

[34] L. Loukas, M. Fergadiotis, I. Androutsopoulos, and P. Malakasiotis. Edgar-corpus: Billions of tokens make the world go round, 2021.

[35] I. E. Maarouf, Y. Mansar, V. Mouilleron, and D. Valsamou-Stanislawski. The FinSim 2020 shared task: Learning semantic representations for the financial domain. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 81–86, Kyoto, Japan, 5 Jan. 2020. -. URL https://www.aclweb.org/anthology/2020.finnlp-1.13.

[36] Y. Mansar, J. Kang, and I. E. Maarouf. *The FinSim-2 2021 Shared Task: Learning Semantic Similarities for the Financial Domain*, page 288–292. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451381.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[38] G. A. Miller. *WordNet: An electronic lexical database*. MIT press, -, 1998.

[39] R. Navigli and P. Velardi. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1134.

[40] N. K. Nguyen, E. Boros, G. Lejeune, A. Doucet, and T. Delahaut. *L3i LBPAM at the FinSim-2 Task: Learning Financial Semantic Similarities with Siamese Transformers*, page 302–306. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451384.

[41] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann. TAXI at SemEval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, California, June 2016. Association for

Computational Linguistics. doi: 10.18653/v1/S16-1206. URL https://aclanthology.org/S16-1206.

[42] Y. Pei and Q. Zhang. Goat at the finsim-2 task: Learning word representations of financial data with customized corpus. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 307–310, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383134. doi: 10.1145/3442442.3451385. URL https://doi.org/10.1145/3442442.3451385.

[43] J. Portisch, M. Hladik, and H. Paulheim. *FinMatcher at FinSim-2: Hypernym Detection in the Financial Services Domain Using Knowledge Graphs*, page 293–297. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451382.

[44] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[45] A. Ritter, S. Soderland, and O. Etzioni. What is this, anyway: Automatic hypernym discovery. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 88–93, -, 2009. -. URL https://www.aaai.org/Papers/Symposia/Spring/2009/SS-09-07/SS09-07-015.pdf.

[46] A. Saini. Anuj at the FinSim task: Anuj@FINSIM¡VLearning semantic representation of financial domain with investopedia. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 93–97, Kyoto, Japan, 5 Jan. 2020. -. URL https://www.aclweb.org/anthology/2020.finnlp-1.15.

[47] K. Shinzato and K. Torisawa. Acquiring hyponymy relations from web documents. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 73–80, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://aclanthology.org/N04-1010.

[48] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pages 1297–1304, -, 2005. -. URL https://proceedings.neurips.cc/paper/2004/file/358aee4cc897452c00244351e4d91f69-Paper.pdf.

[49] T. Stepišnik Perdih, S. Pollak, and B. Škrlj. *JSI at the FinSim-2 Task: Ontology-Augmented Financial Concept Classification*, page 298–301. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450383134. URL https://doi.org/10.1145/3442442.3451383.

[50] Y. Tan, X. Wang, and T. Jia. From syntactic structure to semantic relationship: Hypernym extraction from definitions by recurrent neural networks using the part of speech information. In J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, editors, *The Semantic Web – ISWC 2020*, pages 529–546, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62419-4.

[51] K. Tian and H. Chen. Aiai at the finsim-2 task: Finance domain terms automatic classification via word ontology and embedding. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 320–322, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383134. doi: 10.1145/3442442.3451388. URL https://doi.org/10.1145/3442442.3451388.

[52] E. Tjong Kim Sang. Extracting hypernym pairs from the web. In *Proceedings of the*

45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 165–168, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-2042.

[53] E. Tjong Kim Sang and K. Hofmann. Lexical patterns or dependency patterns: Which is better for hypernym extraction? In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 174–182, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL https://aclanthology.org/W09-1122.

[54] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.