Leveraging Hierarchical Language Models for Aspect-Based Sentiment Analysis on Financial Data

Matteo Lengkeek[a], Finn van der Knaap[a,*], Flavius Frasincar[a]

[a]*Erasmus University Rotterdam, PO Box 1738, 3000 DR, Rotterdam, the Netherlands*

Abstract

Every day millions of news articles and (micro)blogs that contain financial information are posted online. These documents often include insightful financial aspects with associated sentiments. In this paper, we predict financial aspect classes and their corresponding polarities (sentiment) within sentences. We use data from the Financial Question & Answering (FiQA) challenge, more precisely the aspect-based financial sentiment analysis task. We incorporate the hierarchical structure of the data by using the parent aspect class predictions to improve the child aspect class prediction (two-step model). Furthermore, we incorporate model output from the child aspect class prediction when predicting the polarity. We improve the F1 score by 7.6% using the two-step model for aspect classification over direct aspect classification in the test set. Furthermore, we improve the state-of-the-art test F1 score of the original aspect classification challenge from 0.46 to 0.70. The model that incorporates output from the child aspect classification performs up to par in polarity classification with our plain RoBERTa model. In addition, our plain RoBERTa model outperforms all the state-of-the-art models, lowering the MSE score by at least 28% and 33% for the cross-validation set and the test set, respectively.

Keywords: Text data, financial aspect classes, polarity, hierarchical structure of data

1. Introduction

On average, 500 million tweets are sent every day as of 2023, and this number has steadily been increasing over the past years [1]. More generally, the amount of digital text produced by humans is ever-increasing as countries modernize and the world becomes more globalized. This digital text contains rich information interpretable and usable by humans. However, for machines,

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162

Email addresses: matteolengkeek@gmail.com (Matteo Lengkeek), 573834fk@eur.nl (Finn van der Knaap), frasincar@ese.eur.nl (Flavius Frasincar)

the interpretation is harder, as text streams are a lot less structured than number streams. To use this unstructured data, a computer first needs to process text using Natural Language Processing (NLP). Modern statistical tools give researchers the ability to extract information from text and encode it in a quantitative form. These modern tools bring us closer to automated classification of text and extracting associated sentiments.

The focus of this paper is the classification of text using pre-defined financial aspect classes (such as 'Risks' or 'Price Action') and their associated sentiment (on a sentence level). Previous work on NLP in finance has focused primarily on general sentiment in text [2]. A subfield of sentiment analysis is Aspect-Based Sentiment Analysis (ABSA), which provides a fine-grained view of sentiment in text [3]. There are some other works on ABSA in finance but we find that those models treat the aspect classification and polarity (sentiment) classification as separate tasks. We would expect that the aspect classification is taken into account to improve the polarity classification. The FiQA dataset [4] contains multiple level aspect classes from a predefined list from which we use the first (the parent aspect class) and the second level (the child aspect class). Each child aspect class is a more narrowed-down version of the parent aspect class and is, therefore, a subclass. Next to the aspect classes, there is a sentiment that is associated with the child aspect class. The sentence "Auto Trader shares leap in UK's biggest private equity-backed listing" has for example 'IPO' as level 2 aspect class and a positive sentiment (0.641 on a scale of -1 to 1) because the IPO is doing well. We, therefore, observe a hierarchical structure present in the dataset and propose to make use of this by including more upstream (higher in the hierarchy) model data in the more downstream prediction tasks. That is, we aim to incorporate the level 1 aspect classification when predicting the level 2 aspect classification, and incorporate the level 2 aspect classification when determining the sentiment. Furthermore, most finance-based NLP research uses outdated NLP models. In the meantime, the computer science field keeps developing state-of-the-art language models, able to understand context and negations as humans do.

This research, therefore, focuses on using one of the most advanced NLP models named BERT [5] (and its follower version RoBERTa [6]) to classify financial aspects and the associated sentiment for creating hierarchical models. The motivation for these hierarchical models is that we expect that the parent class may help predict the child class since it is a sub-class. We also expect that including the child aspect class when predicting the sentiment is useful because it adds more context to the polarity classification as the sentiment is related to this child aspect class.

This results in the following research question:

Can we improve financial ABSA using hierarchically structured language models combining aspects and sentiment?

In order to answer the previously set question, we propose three hierarchical ABSA models (two for aspect classification and one for sentiment classification) inspired by the LCF-ATEPC model and the FinBERT model, introduced by [7] and [8] respectively. We implement BERT and RoBERTa as model types because they obtain the best performance in many NLP tasks [5, 6].

The main contributions of this paper are as follows. First, we expand the research done on ABSA in finance. Most research has been done on sentiment only, while a good performing ABSA model could add extra possibilities in researching aspect-specific sentiment market reactions. This is useful for policymakers testing their policies on a macro level, but also for investors which want to process market news and posts from microblogs quickly and efficiently. For economic researchers, it is also of value because it can reveal new relations. Second, we introduce new model architectures for datasets that have multiple aspect levels and where the hierarchy of those aspects can be used for polarity classification. Our proposed models are applicable in ABSA task where there are more aspect levels and therefore also of interest outside the finance domain.

The rest of the paper is organized as follows. In Section 2, we discuss previous work related to our research. Then, in Section 3, we present the data used in this study followed by, in Section 4, a description of the proposed methodology. Consequently, in Section 5, we discuss the results obtained using our methods and compare them to other works. Last, Section 6 summarizes the work and suggests future research directions. The source code (in Python) is made publicly available at `https://github.com/mlengkeek/HierarchicalABSA`.

2. Related Work

Various research has been conducted on the topic of NLP (in finance). Some approaches are focused on improving NLP methods and their features, while others are applications of these in the financial domain [9, 10].

Because both the availability of text data and the frontier of methods are expanding rapidly, the importance of text in empirical economics continues to grow [2]. Sentiment from financial text is already used to predict asset price movements [11, 12, 13, 14, 15, 16, 17, 18, 19, 20], future quarterly performances [21], financial risk [22], inflation [2], unemployment [23, 24], and the effects of policy

uncertainty [25]. Next to future predictions, nowcasting (prediction of the present state of economic indicators which usually become available after the period has passed) of macro-economic variables, such as unemployment claims and retail sales, is also performed using NLP methods [23, 26]. Overall, there are many relations found between sentiment and financial measures. Applications of NLP in finance are still quite simple, however, and mostly based on detecting positive, neutral, or negative sentiment.

Another stream in NLP, for example in [3, 27, 28], is focused on ABSA in which the sentiment is extracted with respect to a certain aspect. An example is the sentence "The ice cream is cold". If we take the aspect term 'ice cream', the associated sentiment is positive, since being cold is a positive feature of ice cream. However, if we would replace 'ice cream' with 'soup', the sentiment towards the aspect 'soup' could be negative because soup is often eaten warm. Most ABSA models are based and measured on the Restaurants and Laptop datasets, which contain around 3000 annotated sentences. These datasets originate from the International Workshop on Semantic Evaluation (SemEval), which is a series of international NLP research workshops whose mission is to advance the current state-of-the-art in semantic analysis and to help create high-quality annotated datasets in a range of increasingly challenging problems in natural language semantics. Most of the existing ABSA work focuses on the subtask of aspect term polarity inference and ignores the significance of aspect term extraction (extracting the word, if present, to which the polarity is linked).

[7] proposes a multi-task learning model called Local Content Focus-Aspect Term Extraction Polarity Classification (LCF-ATEPC). Compared to most other ABSA models, this model is capable of extracting aspect terms and inferring aspect term polarity synchronously. By integrating the domain-adapted BERT model, the LCF-ATEPC model achieved the state-of-the-art performance for aspect term extraction and aspect polarity classification in four Chinese review datasets. Besides, the experimental results on the Restaurant and Laptop datasets are better than the state-of-the-art performance on the Aspect Term Extraction (ATE) and Aspect Polarity Classification (APC) subtasks. The LCF-ATEPC model uses its output from the aspect term extraction model when classifying the polarity. The advantage of this method is that it more effectively captures sentiments with regard to context, leading to improved prediction accuracy.

ABSA has been explored for numerous industries but remains quite unexplored in finance. One reason for this is the lack of a well-annotated dataset with aspects focused on the finance domain. A recent release of data for an open challenge called FiQA from the companion proceedings of the 27th World Wide Web Conference (WWW 2018) [4] has provided finance-specific annotations.

The challenge contains an ABSA and Q&A task (task 1 and task 2), and we refer only to task 1 and denote it as FiQA from now on. FiQA contains high-quality labels for aspects (multiple levels) and sentiment. The aspect classes are from a predefined list and are therefore different from traditional ABSA in which an aspect is a term present in the sentence. It is of interest if we can apply similar techniques as LCF-ATEPCS in order to improve both aspect class detection and sentiment analysis. A way to do this would be to take the model output of the aspect classification model and incorporate it for the sentiment analysis.

Exploiting the hierarchical structure of data is not a new concept in ABSA. Most research, however, focus either solely on sentiment detection [29], or do not consider financial data [30, 31, 32]. [33], using the same data, uses the hierarchical structure of the data for aspect classification, ignoring the possibility that aspects contain information regarding the corresponding sentiment. Therefore, using the hierarchical structure of the data for both aspect classification and sentiment classification in the financial domain has, to our knowledge, not been done before.

Furthermore, we find that most studies on NLP in finance have focused on simplistic statistical methods such as bag-of-words or domain lexicons [34]. Human language, however, contains context and negations which are hard to capture by such simplistic statistical methods. In recent years, a lot of advancement in the field of NLP has been made by computer scientists [35]. Current state-of-the-art models are better able to capture negations, discussed aspects, and have some general understanding of language. In addition, we find that the FiQA dataset is relatively small for its number of classes. This makes it hard to train a good NLP model from scratch. Fortunately, a recently introduced model might tackle this problem to some extent.

[5] proposes the Bidirectional Encoder Representations from Transformers (BERT), and it already contains a deep (feed-forward neural network) contextual understanding of language and needs far less task-specific training data to perform well. This language model is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. Bidirectional means that in contrast to previous NLP models, it can read a sentence as a whole instead of from left to right or from right to left. This enables the model to get a better sense of context, especially if words are far away from each other in a sentence. Because the model is already pre-trained on language, it can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, without substantial task-specific architecture modifications. Due to its good understanding of language, it often achieves state-of-the-art results in NLP problems [5]. An application of BERT in finance is

performed by [8] which uses plain vanilla BERT and retrains it on a financial corpus to create a financial BERT sentiment classifier named FinBERT. Building on BERT, [6] introduces RoBERTa, a model with the same architecture but pre-trained differently and on more data (ten times as much). This model performs even better than BERT because of optimized pre-training and a larger dataset. We aim to use both the BERT model with a financial-corpus and the more advanced RoBERTa model in this research.

3. Data

In this study, we use the FiQA dataset which was published by [4] as an open challenge during the WWW 2018 Conference in Lyon, France. The challenge focuses on advancing the state-of-the-art of ABSA and opinion-based Question Answering for the financial domain and consists of task 1 (ABSA) and task 2 (Q&A). We only focus on task 1, so when mentioning FiQA, we mean FiQA task 1 from now on. The data, provided at [36], contains aspect and sentiment information about news headlines extracted from finance domain Web pages like Wikinews, and microblog posts from StockTwits and Reddit. The data contains multiple hierarchical aspect class levels per sentence. We only take the first two levels into account, which is in line with other works and the test set. We denote parent aspect class as level 1, and the more fine-grained child aspect class as level 2. Each text snippet in our dataset has therefore one level 1 aspect and one level 2 aspect. The polarity score is with respect to the level 2 aspect. The dataset has a train and a test part which consist of 1111 and 192 observations, respectively. We use 5-fold cross-validation on the training data and create validation sets for hypertuning. The data splitting approach is explained in more detail in Section 4.

We show an example of an entry of the FiQA dataset in Table 1. The entry should be interpreted as follows: the sentence's aspect class is classified as 'Corporate' (level 1) and within the 'Corporate' class, it is classified as 'Regulatory' (level 2). The sentiment score with respect to 'Regulatory' is 0.549, which is positive, indicating a positive regulatory event.

Table 1: Example entry of FiQA task 1 dataset.

Sentence	AstraZeneca wins FDA approval for key new lung cancer pill
Level 1 aspect	Corporate
Level 2 aspect	Regulatory
Sentiment score	0.549

The frequency distribution (number of occurrences per class) of the predefined aspect classes within the dataset is shown in Table 2. Level 1 aspect classes are more general class descriptions, whereas level 2 aspect classes are specialized descriptions of the level 1 aspect classes. It is clearly seen that there is a class in-balance in both the level 1 and level 2 aspects. Since there is no other financial ABSA dataset publicly available, we accept this and recognize the creation of a more complete financial ABSA dataset as a task for future research. The polarity of the sentences is continuously distributed and ranges from -1 until 1, representing negative and positive sentiment, respectively.

Table 2: Frequency of aspects classes in the FiQA dataset (observation count per aspect class).

Level 1 Aspect	Level 2 Aspect	Frequency of aspect class
Corporate	Appointment	39
	Company Communication	11
	Dividend Policy	40
	Financial	70
	Legal	29
	M&A	75
	Regulatory	17
	Reputation	12
	Risks	55
	Rumors	27
	Sales	91
	Strategy	56
	Technical Analysis	2
Economy	Central Banks	5
	Trade	2
Market	Conditions	3
	Currency	3
	Market	24
	Volatility	11
Stock	Buyside	6
	Coverage	66
	Fundamentals	13
	IPO	8
	Insider Activity	9
	Options	12
	Price Action	496
	Signal	25
	Technical Analysis	96

3.1. Pre-processing

In NLP, it is a common practice to pre-process text before using it in a model. The goal of this pre-processing is to remove characters/words which might cause confusion or only add white noise and therefore worsen the predictions. Popular choices of pre-processing methods for NLP are:

- Making everything lower case;
- Removing numbers;
- Removing punctuation and special characters;
- Removing leading and/or trailing whitespaces.

Because the FiQA dataset consists of news articles and microblogs, we also make use of some specific text pre-processing methods. The news headlines may contain hyperlinks to other pages (e.g., `http://website.com`), so we remove them from the text. Another example is that the microblog posts might include references to other users (e.g., `@username`) or hashtags (e.g., `#hashtag`). We consider this to be noise for a language model, so we strip the sentences from them.

However, if we remove numbers and special characters, we loose economic information if sentences contain financial numbers (e.g., 1 million), dollar/euro signs (e.g., $1M), or stock tickers, which are indicated as a dollar sign followed by three to five characters (e.g., `$ABC`). Therefore it might be wise to try out different configurations of those pre-processing methods and use validation data to find which pre-processing methods have a net positive effect on predictions. We experiment with the pre-processing of casing, numbers, punctuation, and special characters.

3.2. Tokenization

Most NLP models do not process a sentence as a whole. Instead, they tokenize the sentence in tokens. Generally, this means that each word of a sentence is a token and therefore the tokens are split by the spaces in a sentence. Tokenization is also used in BERT models. A graphical example of how such a pre-processing and tokenization process looks like is shown in Figure 1 (using the BERT-base-uncased tokenizer). When classifying with BERT, the `[CLS]` token is added in front of the sentence, indicating that the model's task is (aspect) classification. At the end of each sentence, the token `[SEP]` is added which indicates a separation of sentences. For this research, we use multiple tokenizers, depending on which language model we employ. The process for other tokenizers (BERT-cased and RoBERTa-base) is similar, some differences are in vocabulary, and how they manage sub-words.

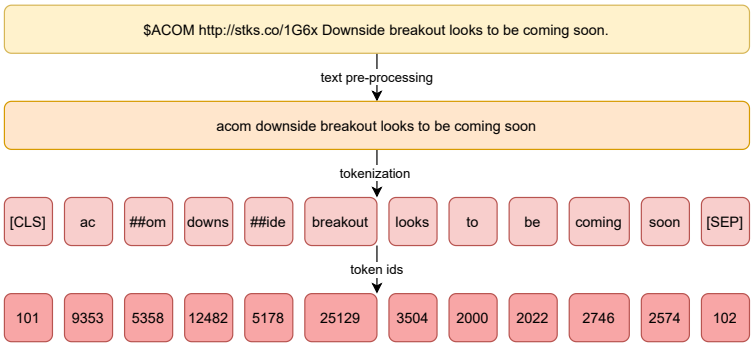


Figure 1: Example of sentence pre-processing and tokenization using the BERT-base-uncased tokenizer.

4. Methodology

As discussed in Section 1, we implement various BERT and RoBERTa models. For both models, we use the hierarchical structure intrinsic to the data in order to improve the predictive accuracy of the models. The models are used for three ABSA tasks: Level 1 aspect classification (L1AC), Level 2 aspect classification (L2AC), and polarity classification (PC).

4.1. BERT & RoBERTa

As model of choice for the FiQA task, we choose BERT and its enhanced version RoBERTa since both have proven to often produce the best results on a variety of NLP tasks. BERT stands for Bidirectional Encoder Representations from Transformers. "Bidirectional" means that in contrast to previous NLP models, this model can capture context of a sentence as a whole. Previous NLP models used to 'read' a sentence from left-to-right or from right-to-left. "Encoder Representations from Transformers" means that the BERT model consists of multiple encoders, encoders being a part of a transformer [37]. For additional details on BERT, we refer the reader to [5].

The implementation of BERT consists of two steps: *pre-training* and *fine-tuning*. In the pre-training phase, the model is trained on unlabeled data using a selection of pre-training tasks. The fine-tuning phase starts with the parameters found in the pre-training phase and then retrains all parameters for the fine-tuning task. This means that for each downstream task the model starts with the same parameters from the pre-training phase but 'fine-tunes' them for the specific task at hand. We use the BERT-base-(un)cased pre-trained model from the huggingface python library. A more finance-specific pre-trained model is provided by [8], which is post-trained (further pre-trained) on the TRC dataset from Reuters (27M words). We also use this domain-specific language model.

[6] builds on the BERT framework and shows that hyperparameter choices for language models have a significant impact on the final results. The authors present a replication study of BERT pre-training, which carefully measures the impact of training data size and key hyperparameters. The authors find that BERT was significantly under-trained and that they can match or exceed its performance by pre-training better. The name their implementation of BERT is: Robustly optimized BERT pre-training approach (RoBERTa). The structure and idea of BERT are not altered, only the way it is trained is modified. The application and fine-tuning are similar to the original BERT model, and therefore the comparison is straightforward.

Since BERT and RoBERTa contain many parameters (over 110M and 125M parameters, respectively), we need to properly hypertune our models. We do so by stratified K-fold cross-validation

with $K = 5$ on the train set. We proceed to use the hyperparameters that perform best in the evaluation sets and retrain on the whole train set to evaluate the model on the test set (which has never entered the training procedure before). We apply stratification because our train set is relatively small compared to the number of classes. We remove classes that occur less than 5 times from our hyperparameter optimization because with $K = 5$, we cannot stratify those classes over the folds. This method is repeated for all of our tasks and models. We take the recommendations of [5] as a starting point of our hyperparameter grid and adjust it for our tasks at hand. The authors also mention that the values of the hyperparameters do not matter as much when the dataset increases to a sufficient size. Since we have a relatively small dataset, we hypertune a broader set of hyperparameters and let cross-validation pick the best set of hyperparameters for each task. We show the hyperparameters ranges we use in Table 3.

Table 3: Hyperparameters grid.

Hyperparameter	**Value(s)**
Dropout probability	0.1
Learning rate	{2e−5, 3e−5, 5e−5}
Batch size	{4, 8, 16, 32, 64}
Training epochs	{2, 3, 4, 5}
Warmup ratio	{No warmup, 0.1}

For our hyperparameter selection, we take the F1 score as a measure for the L1AC and L2AC tasks. The F1 score is a measure of predictive accuracy. The advantage of the F1 score over accuracy is that the F1 score takes class imbalance into account. For the hyperparameter tuning of the aspect classification tasks, we do a regular grid search.

For the PC task, we use the Mean Squared Error (MSE) to determine what the best hyperparameters are. A regular grid search is computationally too expensive (due to the higher complexity of the model used here). We opt for the python package `optuna` [38] which changes the hyperparameters between trials based on the performance of previous hyperparameter trials. The `optuna` package implements the Tree-structured Parzen Estimator (TPE) algorithm, which is a Bayesian way of optimizing hyperparameters (first introduced by [39]). Furthermore, the package prunes trials which do not seem promising. We use `optuna` to find the best hyperparameters for each fold. With the obtained set of best hyperparameters per fold (5 configurations, one for each fold), we test each configuration on the other folds. By averaging the performance over the folds we decide on the ultimate best set of hyperparameters. Not all works report on the original test set because it was released later, they, therefore, report 5- or 10-fold cross-validation results of the training data. For

comparison purposes, we also report cross-validation results. We use 5 fold cross-validation, and take 10% of each fold's training data as validation data. After obtaining the best hyperparameters using the same methodology mentioned previously, we retrain each fold on 100% of the training data and aggregate the cross-validation results by averaging them.

4.2. Optimizer

For our optimizer, we use the AdamW (Adam with weight decay) optimizer as introduced by [40]. [40] provides evidence that the proposed modification decouples the optimal choice of weight decay factor from the setting of the learning rate for both standard SGD and Adam, and improves Adam's performance.

Next to an adaptive optimizer, we use a warm-up ratio for our training. This means that for the first x percent of the total training data, the learning rate is linearly increased from 0 to the intended learning rate. This decision is based on the practices of [8] and of [6] who use warm-up ratios between 0 and 0.1. We hypertune this warm-up ratio parameter as well. [41] shows that warm-up works as a variance reduction technique. The authors find that the adaptive learning rate has an undesirable large variance and can cause the model to converge to suspicious/bad local optima supported by empirical and theoretical evidence.

Next to warm-up, we also experiment with gradual unfreezing, first introduced by [42]. Gradual unfreezing means starting with some layers frozen and during the training process gradually allowing layers to unfreeze and their parameters to be updated. The goal of freezing is to not cause 'catastrophic forgetting' of language. The BERT model has 12 layers with a classification head on top, and we perform trials in which we freeze the first 6 layers and unfreeze them during the training process (after 1-2 epochs), such that the core of the model is not altered at the beginning of training.

We also use freezing for our L2AC model which starts with the transferred L1AC core model. This model uses the BERT model of L1AC as a starting point to become a L2AC model. To fit the task, we need to replace the classification head and randomly initiate some weights. We, therefore, freeze the L1AC BERT layers (all 12) until convergence of the classification head weights and then train the model as a whole to prevent 'catastrophic forgetting'. The huggingface library recommends training all layers at once so this remains our baseline.

4.3. Proposed Hierarchical Structure

The goal of this paper is to investigate if it possible to improve predictive accuracy by making use of the hierarchical structure present in the dataset. To accomplish this, we propose a structure that uses previous models' output or features. That is, we propose to use the parent aspect class to help predict the child aspect class, and to use the predicted child aspect class model output to enhance the polarity classification. We show our intended hierarchical structure in Figure 2. First, in line with the explanation in Section 3.1 and Section 3.2, we pre-process the data, which is shown by the red blocks in Figure 2. Then, we start with L1AC, followed by L2AC conditional on the results from L1AC, which is the right-hand side of the structure in Figure 2. Besides L1AC and L2AC, we also do PC using the output of a plain L2AC model, meaning that the level 2 aspects are not extracted using level 1 aspect information. This is shown by the left-hand side in Figure 2.

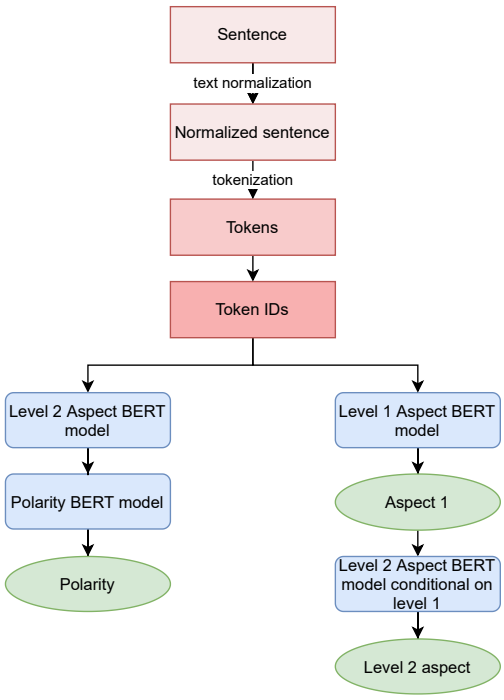


Figure 2: Overview of hierarchical model structure.

4.3.1. Level 1 Aspect Classification

For the first level aspect classification, we make use of classical BERT classifiers, without hierarchical adjustments. Because the dataset is so small, we also make use of the pre-trained finance language model (BERT model weights) from [8], such that we benefit from transfer learning. This language model takes the standard language model and post-trains (further pre-training) on a subset of Reuters' TRC2 (Thomson Reuters Text Research Collection) database, which consists of

1.8M news articles that were published by Reuters between 2008 and 2010. [8] filtered the original database on financial keywords to make the corpus more relevant and in limits with the available computer power. The resulting corpus, TRC2-financial, includes more than 29M words and 400k sentences originating from 46,143 articles. Next to this domain-specific language model, we also try the BERT-base-cased and uncased, and RoBERTa (which is cased) models. To make a classification model of BERT, we take the 768×1 representation of the CLS token and put it through a $768 \times C$ linear layer, where C stands for the number of classes ($C = 4$ for L1AC). Because we deal with multi-class aspects, we make use of the softmax classification head on top of the linear layer. After the probabilities have been assigned to the C classes, we pick the final prediction to be the class with the highest probability.

4.3.2. Level 2 Aspect Classification

For L2AC, we also train a plain model for the following two reasons: as a benchmark, and to use it as extra model input for the PC task. The procedure for the plain model is the same as for L1AC. Besides the plain model, we also train models which make use of the hierarchical nature of the data.

For the plain model, we treat the L2AC as a separate task and train it like it is the only information that we have. The training is similar to L1AC but we now have more specific classes, resulting in an increase in the number of nodes of the linear layer at the end. We, therefore, refer to the methodology explained for L1AC with the difference that $C = 27$ instead of 4.

In contrast to the plain model, we also create models that incorporates information picked up during the L1AC task. We do this in the following two ways:

1. Retraining the L1AC model after replacing the classification head to match the increase in classes (transfer model);

2. Training L2AC models for each of the level 1 aspects and using the L1AC model to select the appropriate L2AC model for the final prediction (two-step model).

We illustrate the process of the first method (transfer model) in Figure 3. We start off with the previously trained L1AC language model which is shown as the light blue box with transformers (Trm) in it on the right. We transfer this language model to our new model and take it as a starting point, which is illustrated with the upper right arrow. This means that we take all of the model parameters of the BERT encoders and their neural networks. We then replace the classification head by initializing a new linear layer with more nodes to match the increase in aspect classes, and

add a softmax classifier (illustrated with the bottom right arrow). The linear layer is initialized with random weights. In order to prevent 'catastrophic forgetting' (when a language model 'forgets' a part of its general language understanding, which is hard to retrain using the fine-tune task), we first freeze the BERT layers and train the classification head until convergence. We do this because the random weights could affect the language model by altering it in an undesirable way through back-propagation of the neural networks. After convergence, we unfreeze the language model and train the model as a whole (illustrated by the most left arrow).

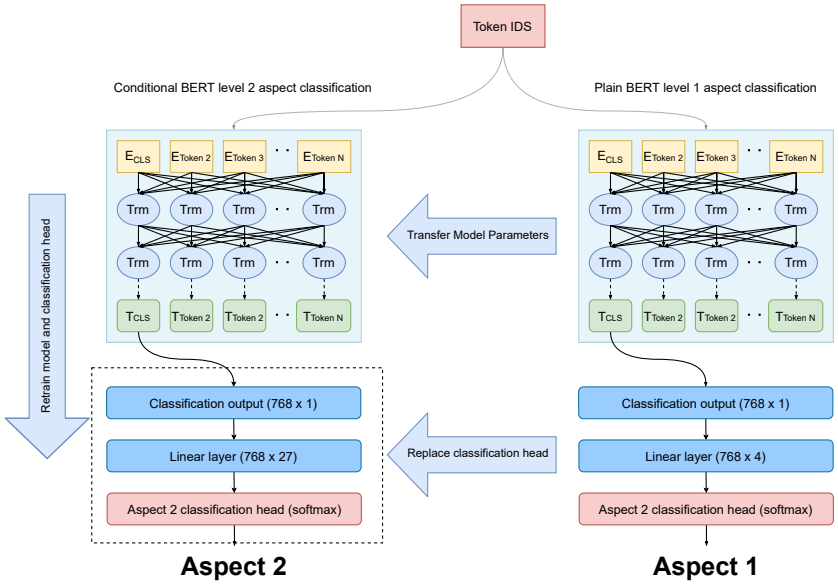


Figure 3: Transfer model process illustrated. Each row of "Trm" represents a Transformer layer with one encoder. The whole BERT model consists of 12 of such transformer layers which each have one encoder.

The other hierarchical implementation is training a model for each of the level 1 aspects. We use the L1AC model to get predictions of which level 1 aspect the sentence belongs to, and use the corresponding level 2 aspect model to get the final L2AC. It should be noted that if the aspect 1 prediction is wrong, the level 2 aspect can never be correct because the level 2 aspect model now predicts within another aspect 1 class. This implies that the theoretical F1 score of L2AC is lower or equal to the F1 of L1AC. The advantage, however, is that each level 2 model within its level 1 aspect is trained with more specific data and with less noise. We assume the level 1 aspects to be known while training the specific level 2 models. We show a visual representation of the two-step model in Figure 4. The right block represents the L1AC model, while the left block represents the specific trained level 2 models (one for each level 1 aspect class). For the final level 2 aspect class prediction, a sentence first passes through the L1AC model. Based on the L1AC model prediction, we pick the

corresponding specific aspect level 2 model as shown by the arrow which goes from the right block to the left block. The eight blue squares followed by four red squares in the bottom left corner represent the classification output, linear layer, and softmax for each of the specific BERT models (three vertically stacked boxes per model which flow into the block that picks the prediction).

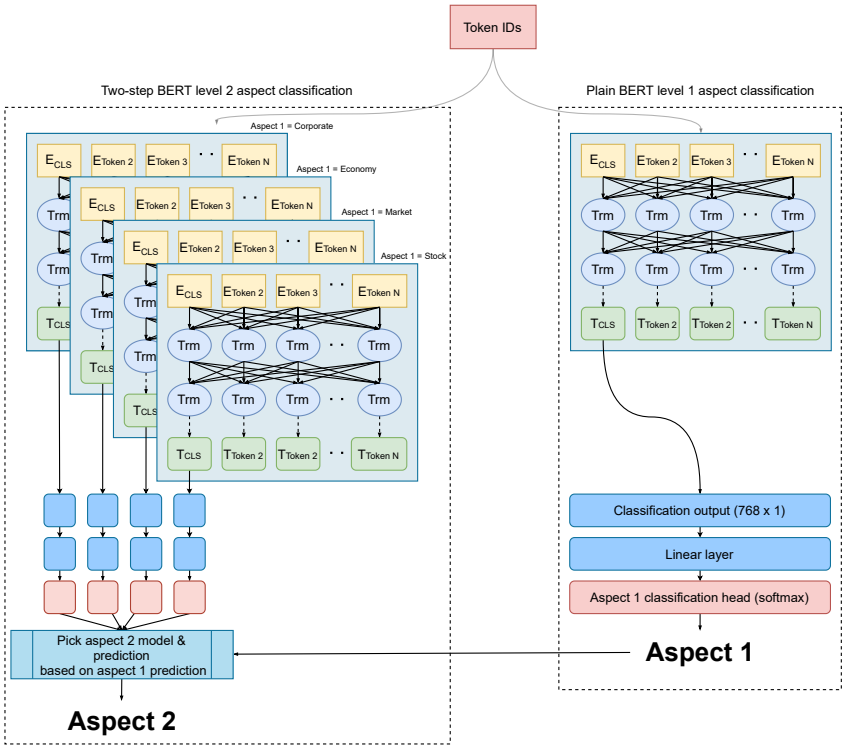


Figure 4: Two-step model process illustrated.

4.3.3. Polarity Classification

For the polarity classification, we again make use of a plain and fusing setup for comparison purposes. The plain models do not take the aspect class prediction into account when predicting sentiment. For the fusing model setup, we concatenate the last layer output of the plain L2AC model with the last layer output of the polarity model (the outputs are of equal size), thereby fusing the model outputs. We expect that by using this configuration, more information about the aspect is taken into account, resulting in a more accurate sentiment prediction.

The motivation and inspiration of this method come from the results of [7], who perform a similar approach but then for aspect term extraction and polarity classification. We do not use the two-step L2AC model outputs for the fusing model because the model outputs are from different models, depending on the level 1 aspect. The language models of the 4 specific level 2 aspect classifiers differ, and therefore, it makes no sense to concatenate output from different models to

the polarity classification output. Other possible hierarchical setups would be to train a sentiment model for each of the level 1 aspects. However, we believe the level 1 aspect to be too general to improve the polarity classification. Another possibility is to train a polarity model for each of the level 2 aspects but this would result in far too many models and hypertuning, especially considering the size of the dataset. We therefore opt for the fusing model as hierarchical polarity model.

In order to allow cross-interactions between the polarity classification output and the concatenated aspect 2 output, we add a dense layer before adding the classification head. A visual representation is shown in Figure 5. The L2AC model is trained independently first. It is represented by the plain model from the L2AC task. After training, we retrieve the classification output of L2AC and add it to the training data for the PC task (shown in the figure by the arrow from the right block to the left block). We use this aggregated data for fine-tune training of the hierarchical PC model. This means that all layers in BERT, the dense and linear layer(s), and the classification head are trained.

The original polarity data ranges continuously from -1 till 1 but we transform this to range from 0 to 1 to work with our model, and transform it back to the original range after the model predictions have been made. We perform this transformation to fit the sigmoid classification head, which gives continuous predictions from 0 to 1.

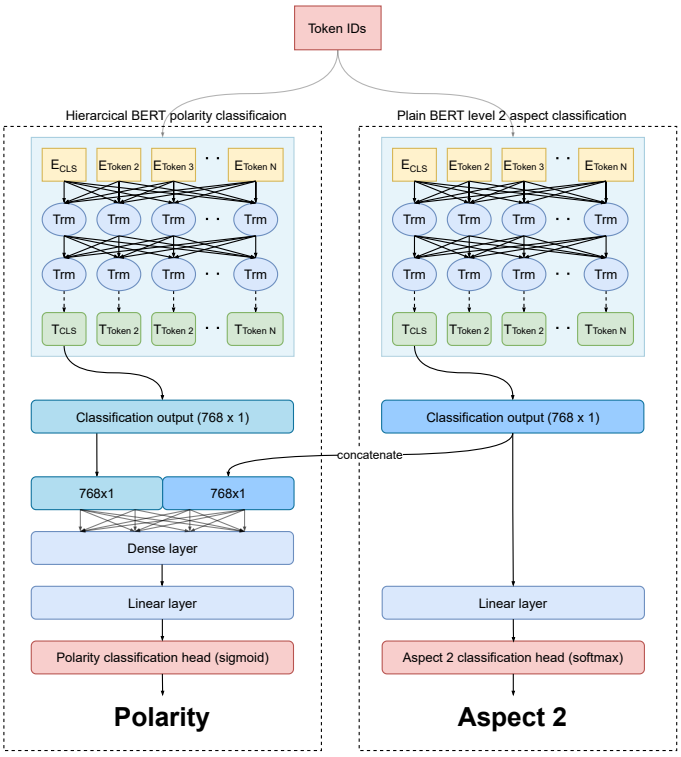


Figure 5: Fusing model process illustrated.

5. Results

We split this section up per task and show the results and discuss them in the order given in Section 4. We first discuss the L1AC task and show the results of hypertuning and the pre-processing of the data presented in Section 3. These results help us with the rest of the tasks. We use the models of the L1AC task and apply them on the data given in Section 3. After discussing the results of the L1AC task, we move on to the L2AC task. We show the hypertuning results and the test results of this task. We also discuss the obtained results and compare them with other models, before moving onto the final task. The same structure applies for the PC task, the last task we discuss in this section.

5.1. Level 1 Aspect Classification

The top of the hierarchy starts with L1AC. We train several models to classify the first level. We create a plain (vanilla) BERT classifier, a BERT classifier trained on a financial corpus, and the recently introduced RoBERTa classifier. Furthermore, we try several text normalization setups. For our text normalization methods, we try removing dollar signs, removing numbers, removing punctuation, and lower-casing the text. We test the possible combinations of the setups for aspect level 1 and continue with the optimal setup for the more downstream tasks (with testing to confirm if findings still hold for downstream tasks).

5.1.1. Hypertuning & Text Normalization

For the L1AC task, we compare different text normalization strategies and tokenizers. We first discuss the hypertuning results. Table 4 shows the set of hyperparameters which work best for L1AC in the 5-fold cross-validation train set. We find that the more refined models RoBERTa and BERT-TRC2 have a slower learning rate than the BERT base models. An explanation for this is that the TRC2 model already has a better financial pre-training and that RoBERTa in general has a better pre-training. This implies that they have to learn less and therefore can achieve similar/better results with a slower learning rate. The number of epochs, however, is the same for all models.

Table 4: Best hyperparameters for 5-fold cross-validation on the training data using optimal text normalization techniques.

Aspect 1 Model	Learning rate	Epochs	Batch size	Warm-up ratio	Validation F1
BERT-base-uncased	5e-5	5	4	0.1	0.895
BERT-base-cased	5e-5	5	8	0	0.871
BERT-TRC2	3e-5	5	4	0	**0.897**
RoBERTa-base	3e-5	5	8	0	0.893

Furthermore, we observe that the uncased models use smaller batch sizes. One possible reason is that the uncased language models require smaller batch sizes to learn to recognize small specific signals such as stock tickers, which are more easily recognized with casing. In bigger batch sizes, it is harder to pick up such small details because the iteration contains more sentences and the iteration's learning is averaged over all sentences in the iteration. Furthermore, we find that in the validation set BERT-TRC2 achieves the best performance, followed by BERT-base-uncased. It is surprising that RoBERTa performs worse than BERT-base-uncased, but the difference is small.
Next to hypertuning, we also experiment with text normalization techniques. We show the results of the text normalization ablation experiment in Table 5 using F1-scores. We find that keeping dollar signs and removing punctuation and numbers from the sentences works best across the 5 cross-validation folds. This result holds for all our BERT and RoBERTa models, although the effect is stronger for the BERT models.

Table 5: Effects on F1-score of different text normalization strategies on 5-fold cross-validation on the training data.

Aspect 1 Model	Effect of stripping dollar signs	Effect of stripping numbers	Effect of stripping punctuation
BERT-base-uncased	-0.020	+0.017	+0.027
BERT-base-cased	-0.008	+0.007	+0.015
BERT-TRC2	-0.033	+0.010	+0.004
RoBERTa-base	-0.001	+0.011	+0.009

Taking all these findings into account, we assume that similar adequate normalization strategies hold for the L2AC and PC tasks. We again perform A-B testing and indeed find that these normalization strategies also work well for the more downstream tasks. We, therefore, remove numbers and punctuation but keep dollar signs for all tasks.

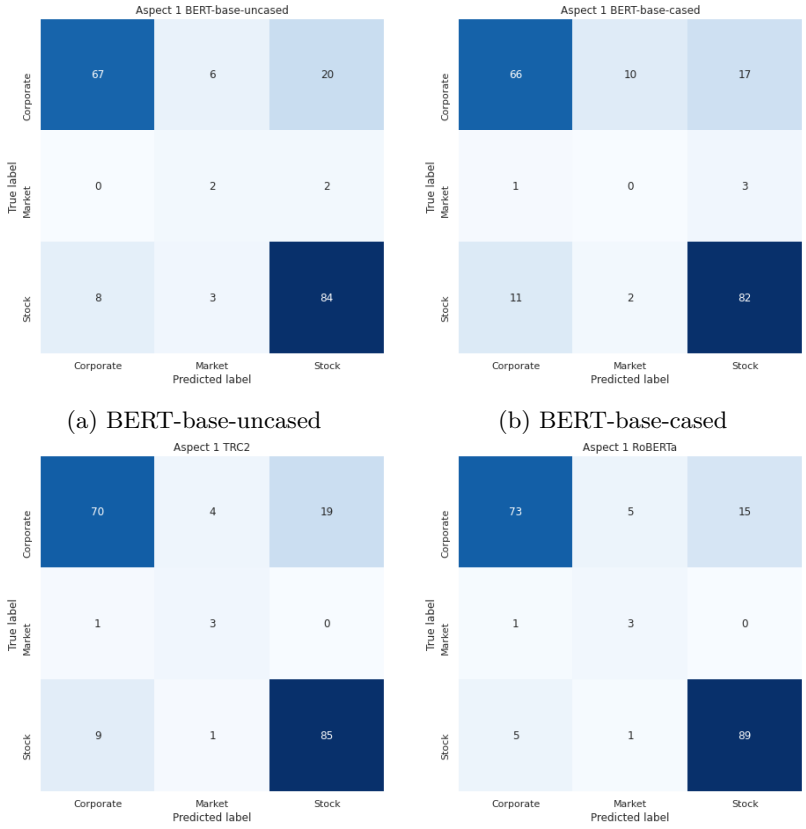
5.1.2. Level 1 Aspect Classification Test Results

We use the found optimal text normalization techniques and put the model to the test in the test set. The test results of L1AC are shown in Table 6. While comparing the language models, we find a slight improvement from BERT-base-uncased to BERT-TRC2. [8] finds similar results, more specifically around a 2% increase when using the domain trained model.

Table 6: F1 scores of the various BERT and RoBERTa model(s) on Level 1 aspect classification

Aspect 1 model	F1-score
BERT-base-uncased	0.81
BERT-base-cased	0.79
BERT-TRC2	0.83
RoBERTa-base	**0.86**

We find F1 scores of 0.81 and 0.83 for BERT-base-uncased and BERT-TRC2, respectively, which is a 2.5% increase. This result confirms the benefits of pre-training BERT models on a task-specific language domain, which is expected to work better than a general language domain. Interestingly, the RoBERTa model performs better than the TRC2 model in the test set in contrast to the results of the validation sets, which implies that the better pre-training or RoBERTa outweighs the domain-specific post-training of TRC2. The differences in confusion matrices are shown in Figure 6. The confusion matrix is 3x3 instead of 4x4 as the test set does not include any 'Economy' entries and because our models never incorrectly predict 'Economy'.



(a) BERT-base-uncased (b) BERT-base-cased

(c) BERT-TRC2 (d) RoBERTa-base

Figure 6: Aspect 1 test data confusion matrices.

We observe that the RoBERTa and TRC2 models have relatively higher diagonal (top left to down right) values than the base models. Another observation that stands out from the confusion matrices is the confusion between 'Stock' and 'Corporate', especially when the true label is 'Corporate' and 'Stock' is predicted. This confusion is slightly lower for the TRC2 and RoBERTa models, which comes back in the F1 scores. We use the RoBERTa model for further hierarchical model setups and include the best performing BERT model (TRC2) as a reference for the performance of BERT.

5.2. Level 2 Aspect Classification

Building on the results found for L1AC, we continue to L2AC and create 2 base models (BERT-TRC2 to represent BERT, and RoBERTa) and 2 hierarchical models (two-step RoBERTa and L1AC transfer model RoBERTa). Hypertuning is done all over again and text normalization strategies are the same as for L1AC since they proved to work best for L2AC. The first observation is that the model which uses the transferred model weights from an L1AC model results in poor performance. None of the freezing techniques result in a sufficient performance and all transfer models result in F1 scores ranging around 0.2, while other models result in F1 scores around 0.7. We, therefore, do not include the transfer model in confusion matrices and performance tables.

As an explanation for this relative bad performance, we suspect the model to adjust its parameters too much to the task at hand (L1AC at first) and since this is a simpler task, it suffers from 'catastrophic forgetting' because it gets rid of signals which are noise for L1AC but useful for L2AC. Especially since the model goes from 4 to 27 classes, we believe that too much info is lost to create a sensible L2AC model. We, therefore, do not advise retraining fine-tuned models, but to start and train from the pre-trained models instead.

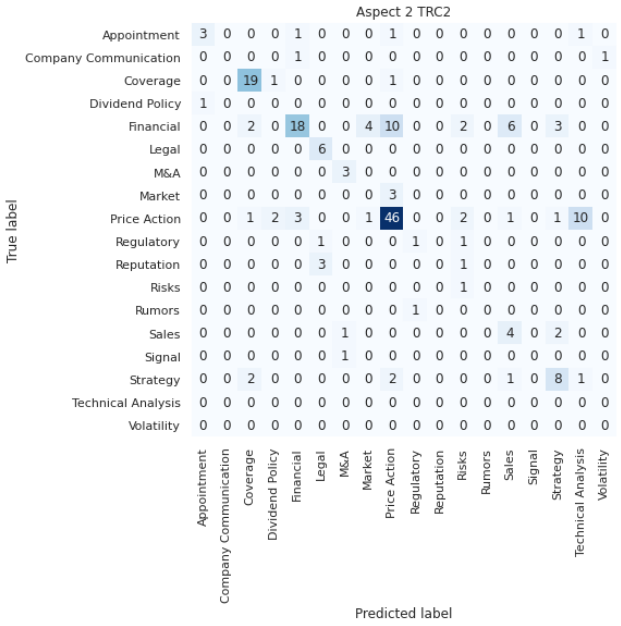
We show the optimal hyperparameter for our other L2AC models in Table 7. We find that in the validation set the hierarchical two-step model performs best. In terms of epochs and batch size, all models are similar (RoBeRTa-base needs a slightly smaller batch size). The RoBERTa models perform best with warm-up, while the BERT-TC2 model performs best without. We use these hyperparameters for our test set.

Table 7: Optimal hyperparameters for Level 2 aspect classification.

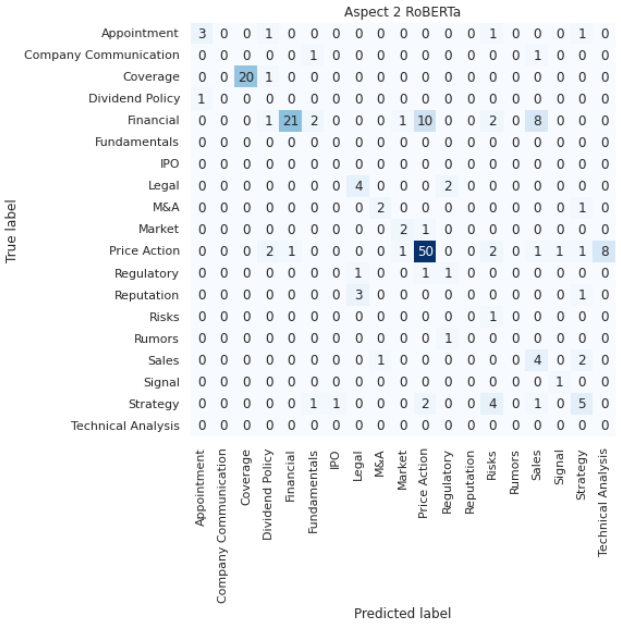
Level 2 Aspect model	Learning rate	Epochs	Batch size	Warm-up ratio	Validation F1
BERT-TRC2	5e-5	5	8	0	0.691
RoBERTa-base	3e-5	5	4	0.1	0.741
RoBERTa two-step	5e-5	5	8	0.1	0.753

Using the hyperparameters found, we obtain the test results. The confusion matrices of the test results are shown in Figure 7. The first observation is that all three models perform quite well. We do see improvement of RoBERTa over BERT by more diagonal entries, which is in line with L1AC findings. Another observation is that both RoBERTa models make fewer mistakes compared to BERT. We specifically see a lot of mix between 'Financial', 'Price Action', and 'Sales', and between 'Price Action' and 'Technical Analysis', and between 'Strategy' and 'Sales'. We understand the confusion since the aspect class 'Financial' is quite broad and can also cover the other classes 'Price

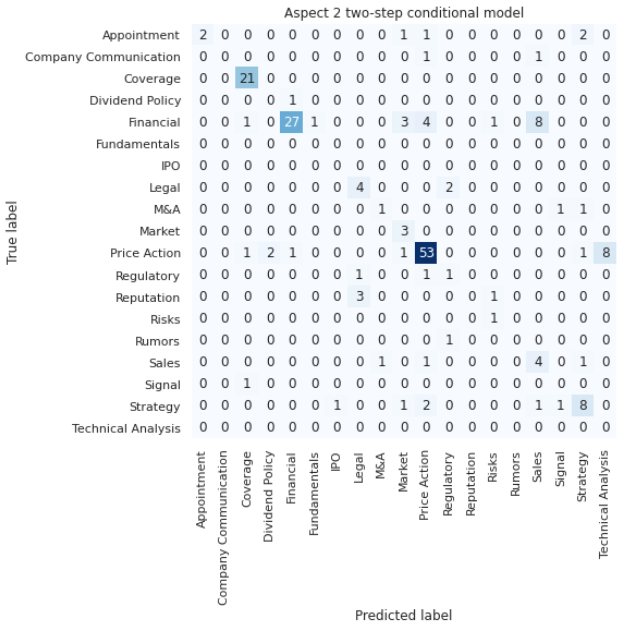
Action' and 'Sales'. We do, however, see that the RoBERTa models have a higher recall than the BERT model. A possible explanation could be that the longer pre-training of RoBERTa gives it a better understanding of language, which enables it to better distinguish the subtle differences between the classes and therefore yield better results.



(a) BERT-TRC2



(b) RoBERTa-base



(c) RoBERTa two-step

Figure 7: Aspect 2 confusion matrices

Another observation is that the two-step RoBERTa model has less mix-up between 'Financial' and 'Price Action' compared to the base RoBERTa model. This can be explained by the hierarchical structure since both classes originate from a different L1A class. This means that if the L1AC model predicts the L1A correctly, the two-step RoBERTa can never pick a class outside of the L1A and hence can not make this confusion. This is an advantage of the two-step model as long as the first step prediction is quite good, which is the case for L1AC (F1 scores around 0.9). This basically means that the two-step model decreases the possible wrong classes and narrows down the scope in which it predicts.

The downside is also clear, if the L1AC is wrong, it never predicts the L2A correctly. However, since the L1AC performance is good, the advantages out-weight the disadvantages in terms of F1 score.

There is also a second advantage, namely because of this narrowing down, the models which are trained for L2AC are trained on a more specific task and could therefore possibly capture nuances better. We observe this phenomenon for example for 'Strategy' (L1A: 'Corporate'), where the two-step model is better in recalling 'Strategy' while the RoBERTa model picks other classes belonging to the same L1A more often (e.g., confuses it with 'Risks'). The two-step model has an advantage because it has specialized L2A "expert" models. Overall, if we look at the F1 scores, we conclude that the RoBERTa models outperform the BERT model and that the two-step model takes advantage of the hierarchical structure to outperform the plain RoBERTa model.

5.2.1. Level 2 Aspect Classification Test Results

We have also collected L2AC results from other works reporting on the FiQA challenge. During the writing of these works, the test set was not yet published and the other works, therefore, report using 5-fold cross-validation (5-CV) on the train data. We prefer to use the published test set because in this way we have more training data and because it implies the exact same test set for everyone, but for comparison purposes, we also reported the 5-CV results.

Table 8 shows the results from other works on top and our results below the dotted line. It should be noted that the cross-validation test results are self-reported and depend on the way the splits are made. We assume this effect to even out over the splits, since at the end the same data is tested, so we also compare the cross-validation results. Furthermore, not all papers use the same number of folds. If the folds differ from 5 we denote it between brackets after the model name.

The first observation is that our BERT and RoBERTa model(s) perform relatively well. Out of our models, the two-step model performs best. Our two-step model outperforms the plain RoBERTa model in both the CV and test set.

When we look at the other works that report on the original test data, our two-step model performs best, followed by our other proposed models. When we compare our result to the BERT model of [43], we observe a big difference. Explanations in why our model performs better could be that we use a language model trained on a financial corpus (TRC2), or that we use the more robust RoBERTa. Both explanations make sense if we look at the L1AC results. Another possible explanation could be that we use an extensive hyperparameter grid which we optimize using the Tree Parzen Estimator, resulting in better performance, while [43] does not seem to do hypertuning.

Table 8: F1-scores of the various proposed RoBERTa/BERT models on Level 2 aspect classification compared to other works (our models are on the bottom and separated by the dotted line).

Aspect 2 model	5-CV train data F1-score	Test data F1-score
CUKG-Tongji (FiQA)	n.a.	0.32
IIT-Delhi (FiQA)	n.a.	0.02
Simple ELMo mean pooled [33]	0.64	n.a.
Pre-trained LM w/ wikitext [33]	0.66	n.a.
Fine-tuned LM w/ VIC gradual unfreezing [33]	0.65	n.a.
Supervised Classifier w/ VIC long/short [33]	0.69	n.a.
Supervised Classifier w/ VIC Aspect 1 [33]	**0.75**	n.a.
bidirectional LSTM RNN [44] (10-CV)	0.69	n.a.
INF-UFG [45] (10-CV)	0.54	n.a.
BERT [43]	n.a.	0.46
Deep-FASP [46]	0.65	n.a.
BERT-TRC2	0.68	0.60
RoBERTa-base	0.70	0.65
RoBERTa-base two-step	0.72	**0.70**

In terms of the 5-CV training data, we see that the performance is higher for our proposed models, which could be explained by the slightly different ratio of microblogs to news headlines sentences in the test set. In terms of performance, we find that the 'Supervised Classifier w/ VIC Aspect 1' performs best, followed by our two-step model. Interesting to see is that both models make use of aspect 1, strengthening the hypothesis that making use of hierarchical structure in ABSA is fruitful. It should be noted that the two-step model only makes use of the available train data, while the VIC model is fine-tuned on a financial corpus (like TRC2). Following findings of L1AC, we can imagine the two-step model to improve its current F1 score when the RoBERTa model would be pre-trained on a financial corpus instead of a general one.

Last, we conclude that refitting a classification head and retraining does not work for RoBERTa and that the two-step RoBERTa model performs better than the plain RoBERTa model. Furthermore, both RoBERTa models perform better than the best BERT model, similar to L1AC. Because the F1-score of L1AC is sufficiently high, the benefit of specialized models creates better predictions than a direct L2AC model. Using this two-step model, we achieve the best result in the test set and the second best result in the 5-CV set, only outperformed by a model which has more domain-specific pre-training data.

5.3. Polarity Classification

Now that both aspects have been classified, we discuss the results of the polarity classification. We again use the TRC2 model as the best BERT model reference and the plain RoBERTa as RoBERTa reference because of their performance in the previous tasks. The fusing model is built on RoBERTa since it produces the best results for both L1AC and L2AC. As extra input for the fusing model, we use the plain L2AC model because the two-step output layers are not from the same model and therefore would not make sense to combine in the fusing model. We first discuss the hypertuning results of the fusing model, followed by the test results.

5.3.1. Hypertuning

Because we add a dense layer for the fusing model, we need to hypertune the size of this layer as well. We show the validation results of the hypertune process in Table 9. As a first observation, we find that all optimal models perform relatively similarly with a MSE of around 0.05. The TRC2 model has slightly worse performance than the RoBERTa models, a pattern that we notice in all tasks. The MSE of the fusing models and the plain RoBERTa model are similar, and we do not see huge improvements of MSE when we use the fusing model.

It should be noted that each of the fusing models could theoretically obtain the same performance as the plain RoBERTa model by setting the weights of the L2AC input to zero. This means that the fusing model is basically a super-set of the plain model and theoretically could always perform the same or better, given sufficient training data. This is not always the case in practice as a neural network converges to a local optimum.

In terms of hyperparameters, we see a slower learning rate for all fusing models compared to the plain models. An explanation is that the fusing models already contain a lot of information of the context and aspect class, and the model, therefore, needs to learn less to finish the polarity classification task. This explanation implies that the aspect is correctly taken into account by the

fusing model. The difference of performance between the 'no dense layer' model and the fusing models with dense layer is small, which means that the addition of a dense layer does not really allow for that much more interaction and complicated relations between sentiment and aspect.

Furthermore, we find that the fusing model with a dense layer of size 96 has the best performance during hyperparameter optimization. Because the model with a dense layer of size 96 yields the lowest MSE during hypertuning we pick this size for the test set, where we can test if the fusing models performs better than the plain model.

Table 9: Hyperparameter validation results of the BERT / RoBERTa models. Best validation MSE (lowest) is highlighted in bold.

Sentiment model	Learning rate	Epochs	Batch size	Warm-up ratio	MSE
BERT-TRC2	5e-5	5	16	0.1	0.0586
RoBERTa-base	5e-5	4	32	0	0.0543
Fusing RoBERTa - no dense layer	3e-5	4	8	0	0.0534
Fusing RoBERTa - dense layer size 12	2e-5	5	4	0.1	0.0532
Fusing RoBERTa - dense layer size 24	3e-5	5	16	0.1	0.0548
Fusing RoBERTa - dense layer size 48	2e-5	4	8	0	0.0582
Fusing RoBERTa - dense layer size 96	2e-5	5	8	0.1	**0.0519**
Fusing RoBERTa - dense layer size 192	2e-5	5	16	0.1	0.0529
Fusing RoBERTa - dense layer size 384	2e-5	4	8	0	0.0544
Fusing RoBERTa - dense layer size 768	3e-5	5	16	0	0.0526

5.3.2. Polarity Classification Test Results

We show the test results in Table 10. The other models and our proposed models are separated by the dotted line. Overall, we observe that BERT and RoBERTa models perform best on this task, namely FinBERT and our own three proposed models.

Table 10: MSE scores of the various BERT models on sentiment extraction compared to published papers.

Sentiment model	5-CV data MSE	5-CV data R^2	Test data MSE	Test data R^2
FinBERT [8] (10-CV)	0.07	0.55	n.a.	n.a.
Simple ELMo mean pooled [33]	0.13	0.17	n.a.	n.a.
Pre-trained LM w/ wikitext [33]	0.09	0.32	n.a.	n.a.
Fine-tuned LM w/ VIC gradual unfreezing [33]	0.09	0.38	n.a.	n.a.
Supervised Classifier w/ VIC long/short [33]	0.08	0.40	n.a.	n.a.
Deep FASP [46]	0.09	0.41	n.a.	n.a.
Multi-Channel CNN [44] (10-CV)	0.11	0.29	n.a.	n.a.
Support Vector Regressor [45] (10-CV)	0.16	0.17	n.a.	n.a.
Linear Support Vector Regressor [43]	n.a.	n.a.	0.25	n.a.
CUKG-Tonghji (FiQA)	n.a.	n.a.	0.12	0.29
IIT-Delhi (FiQA)	n.a.	n.a.	0.15	0.13
BERT-TRC2	0.06	0.61	0.10	0.40
RoBERTa	**0.05**	**0.68**	**0.08**	**0.54**
Fusing RoBERTa - dense layer (size 96)	0.05	0.66	0.09	0.51

We again see an advantage of using the RoBERTa model over the BERT model. In terms of MSE, the scores are close but if we take R^2 as measure, we observe that both the fusing and the plain RoBERTa have substantial higher R^2 scores than the BERT-TRC2 model.

The difference between BERT-TRC2 and FinBERT however is unexpected (BERT-TRC2 being better than FinBERT). The BERT-TRC2 model uses the language model (TRC2) introduced by [8] and is also based on the BERT platform. We, therefore, expected similar performance. Furthermore, we use 5-fold cross-validation like most works, while FinBERT uses 10-fold cross-validation. This means that we also have less training data per fold, making it even more surprising. We find multiple explanations. The first one is that we use an advanced hyperparameter optimizer (`optuna`) which enables us to search a broad grid. Furthermore, we use AdamW [40], a decoupled weight decay version of Adam. The weight decay prevents overfitting and ensures better out-of-sample performance. Last, we use text normalization techniques which work specifically well for this financial dataset.

We hypothesize that the finance-specific language model helps more with aspect classification since it learns the small nuances between financial aspect classes required to classify correctly better. For polarity classification, however, there is also such a thing as general sentiment of words such as "positive", "good", "excellent", which are positive in finance as well. We, therefore, think that the benefit of using a domain-specific language model is smaller than for aspect classification. This does not mean that the domain-specific language model is not necessary because we still believe it can predict the sentiment with respect to the aspect class better as it has more financial context knowledge (assuming that a large amount of financial data is available).

To examine the previous hypothesis, we perform an ablation experiment with our plain BERT model and BERT-TRC2 model, where the only difference is that one uses a general language model and the other a financial-domain language model. We find results in line with the results of [33] and our hypothesis. The plain language model achieves a very similar MSE (0.098 and 0.097 for BERT-base-uncased and BERT-TRC2, respectively), while in the aspect classification tasks we find a bigger difference in performance in favor of BERT-TRC2.

The model which performs worst (both in the 5-CV train dataset and in the original test dataset) is the Support Vector Regression. This does not come as a surprise since it is the simplest model in terms of NLP advancement. Generally, we see an increase in performance the more advanced the model is, a pattern which we also observe in the aspect classification tasks.

The models published in the FiQA challenge [4] that report on the test set perform relatively

bad compared with our models in terms of MSE, but especially in terms of R^2. We cannot explain this difference since no information is given on which models the contestants use. The difference in MSE and R^2 between the 5-CV train set and original test set is, however, something which we can explain. We observe relatively higher MSE scores in the original test set accompanied by relatively high R^2 scores for their MSE score. When we investigate the test sets, we find that the variance of the sentiments is higher in the original test set than in the train set (from which the 5-CV set is created).

Furthermore, the original test set contains relatively more sentences coming from microblogs (relative to the number of sentences coming from news headlines). The sentences from microblogs often use short sentences, jargon, and short opinions in contrast to the news headlines which are more formal and better structured. Combining these observations, we conclude that the composition of the original test set makes it harder to predict correctly, resulting in higher MSE scores. Given the higher variance in the original test data, we can also explain why the R^2 scores are higher given the same MSE score. Because R^2 measures the portion of explained variance you can still get a high R^2 with a high MSE as long as a high portion of variance is explained.

The goal of this paper is to examine the possibility of exploiting the hierarchical structure of the dataset and incorporating the aspect classification in the polarity classification. We incorporate the aspect classification by using the model output of our RoBERTa L2AC model in our proposed RoBERTa fusing polarity classification model. We assume that incorporating the aspect class improves the polarity classification.

During validation, we find evidence of the above hypothesis because the fusing model outperforms the plain RoBERTa model. In the test and 5-CV train set, however, we find that the plain model slightly outperforms the fusing model. This implies over-training in the hypertuning phase for the fusing model and that the plain model actually performs better. Because the fusing model has more parameters than the plain model, it is more prone to overfitting. We have two explanations for why the fusing model does not perform better.

One explanation is that the plain model already takes sufficient aspect classification information into account. This would mean that the addition of the L2AC model output does not help to classify the polarity better.

Our other explanation is that there is too little data to learn how to incorporate the aspect classification data sufficiently enough to improve predictive power. We observe in terms of hyperparameters that the optimal set for the fusing model uses a slower learning rate. Given its similar

performance, this implies that the model gains information of the concatenated model output. However, in terms of performance, we do not notice an improvement in both 5-CV train set and the original test set. The concatenation of the L2AC output creates a lot of extra parameters which need to be optimized during training. It could be that not all of the output is useful for polarity classification. The addition of the aspect model output, therefore, contains not only signal (about the aspect class) but perhaps also noise which makes the training procedure harder. We use approximately 1100 sentences to train the L2AC model and PC model. If there is more data we expect the L2AC to be better, and the cross-interaction of the concatenated model output to be exploited more efficiently. We need a bigger dataset to test this hypothesis, however.

6. Conclusion

In this paper, we focus on language model based approaches for ABSA on a sentence level. We present a summary of the main findings and implications in Section 6.1. In Section 6.2, we present limitations of our research and suggestions for future work.

6.1. Practical and Theoretical Contributions

This paper extends the work of [8], who proposed FinBERT, a sentiment model for financial sentences. We build on this idea by implementing an Aspect-Based Sentiment Analysis model on the FiQA dataset, which was published at the WWW 2018 Conference. This dataset contains multiple level aspect classes (parent and child), which motivates the use of the observed hierarchical structure of the data for aspect and sentiment classification, as parent classes could contain valuable information about child classes, and child classes about the corresponding sentiment. The main practical contribution of this paper is that we investigated whether we can improve the final aspect classification and the corresponding polarity using hierarchical models. We aimed to achieve this by proposing several hierarchical BERT/RoBERTa models which use parent aspect classes when predicting child aspect classes, and child aspect classes when predicting polarity. We have shown that the final aspect classification can be improved using hierarchical models, beating all the state-of-the-art models.

Second, we investigated the performance of different text normalization techniques for financial data. We found that the standard procedure of removing punctuation (next to hyperlinks, hashtags, and usernames) improved the predictive accuracy. As we deal with financial sentences, we examined the effect of removing dollars signs and numbers separately. We found a positive effect in predictive

power when we removed numbers, even though BERT contains some form of numeracy, and therefore concluded that numbers add more noise than signal. When we removed dollar signs performance worsened. We, therefore, concluded that keeping dollar signs is beneficial for financial ABSA tasks.

Third, in the parent aspect classification task (L1AC), we observed that RoBERTa models perfrom better than BERT. Furthermore, in line with expectations, we found that domain-specific pre-trained language models outperform general language models. The benefit of using a domain-specific BERT model is, however, less than using a general RoBERTa model. We, therefore, recommend using RoBERTa over BERT, and to use a domain-specific RoBERTa language models when available.

Fourth, For the child aspect classification task (L2AC), we introduced two new architectures which make use of the hierarchical structure of the data. The first architecture, the transfer model, is a model which is first trained on the parent class and then adjusted and retrained for the child class. This transfer model did not yield good results because of 'catastrophic forgetting'. The other model architecture which we introduced is the two-step model, which consists of a model to predict the parent class and a model for each of the parent classes to predict the corresponding child class.

The two-step approach resulted in better predictions than training a model to predict the child class directly. We outperformed the plain RoBERTa model significantly in the test set. Furthermore, the two-step model achieved state-of-the-art results compared to other published works. We achieved the best F1 score on the test set, and achieved the second-best score on the self-reported cross-validation test set. When we excluded models which have been pre-trained on other financial text and only compared models which use the same training data, we also achieved the best performance in the cross-validation test set. The two-step model, therefore, uses the hierarchical structure of the data efficiently and improves predictions.

Last, we also proposed a new architecture for polarity classification. We introduced a model which uses the model output of the L2AC and concatenates it with the model output of a RoBERTa model, which is trained for polarity classification. We placed a small neural network on top of the two concatenated RoBERTa model outputs and used this to predict the sentiment. We found evidence that the model incorporates the concatenated output because the model needs a slower learning rate as it already gets part of its information from the L2AC model output. In terms of performance, however, the results are up to par with the plain RoBERTa model. We attribute this to the relatively small train set which is used to fine-tune the RoBERTa model, while simultaneously training a neural network on top. We hypothesize that with sufficient data the fusing model

could work better than the plain RoBERTa. Furthermore, our models achieved state-of-the-art results in both the original test set and the cross-validation test set. We attribute this to our text normalization, hypertuning, and optimization techniques combined with the advanced RoBERTa model.

Our models can be used for various financial applications. Most research has been done on sentiment only, whereas a good performing ABSA model could be useful for various financial tasks. First, our proposed models could add extra possibilities in researching aspect-specific sentiment market reactions, as we show that using the hierarchical structure of data could be beneficial for classification, which could result in more accurate predictions. Furthermore, our proposed models could be used for nowcasting, achieving more accurate predictions. They could also be used for future predictions, like inflation and unemployment. Having accurate forecasts of macro-economic variables is extremely important, as it provides valuable information for policymakers.

6.2. Research Limitations and Future Work

Next to the findings of our research, we also have some points of discussion. The first point of discussion is the lack of data. The dataset is quite small, containing only 1303 entries, especially considering the 27 aspects level 2 classes which are not uniformly distributed over the data. Because no other financial ABSA database is publicly available this is the best we can do at the moment but it would be interesting to investigate if the same results hold with bigger datasets. A bigger financial ABSA dataset can also improve the financial ABSA models which makes them more valuable for economists who use such financial ABSA models to create forecasting models or decision models.

Another point of discussion is the test set. Only a few works report on the original test set because their work was released before the release date of the original test set. This means that the associated papers report on test sets created using cross-validation from the train set. It would be better if all works reported on the exact same test set. Although the test set from cross-validation may change, we do not think that the results would differ much but it would make for a more fair comparison.

This work can be used as a starting point for future research as well. We have various suggestions. First of all, a bigger financial ABSA dataset could be created, or the FiQA dataset could be expanded. This bigger dataset can train more complex and better models, for example, the hierarchical polarity classification model. Next to the dataset, it would also be useful if a financial domain-specific RoBERTa language model is created, the benefits are clear for BERT and we expect the same

increase in performance for RoBERTa. One can also use our models for automatically classifying financial aspect class and polarity of sentences from news headlines or microblogs. Using the classifications, one could follow market trends, find new relations, or create financial indicators.

Another interesting area for future research is to investigate if there is a way to take the financial numbers explicitly into account, instead of discarding them or regarding them as text because we had to disregard them for performance reasons. A possible suggestion is to use named entity recognition to find financial figures in a sentence, extract them, and add them as a numeric variable.

Last, we have shown that a two-step model worked best in our case but we can imagine it working less well when the first step prediction is poor. An interesting take on this would be to create an ensemble of the direct model and the two-step model, which gives weights to both models prediction based on the performance of the first step prediction. Another approach of deciding the weights would be to take the certainty of the level 1 prediction (of the two-step model) into account, taking the two-step prediction when it is certain (e.g., more than 80% sure of L1AC) and the direct L2AC when it is not certain. A ridge regression could also be used to decide which weights to use to combine the predictions.

References

[1] Internetlivestats, 2023. URL: `https://www.internetlivestats.com/twitter-statistics/`.

[2] M. Gentzkow, B. Kelly, M. Taddy, Text as data, Journal of Economic Literature 57 (2019) 535–574.

[3] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (2015) 813–830.

[4] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, A. Balahur, WWW'18 open challenge: Financial opinion mining and question answering, in: Companion Proceedings of the Web Conference 2018, ACM, 2018, pp. 1941–1942.

[5] J. Devlin, M. W. Chang, K. Lee, 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, ACL, 2019, pp. 4171–4186.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[7] H. Yang, B. Zeng, J. Yang, Y. Song, R. Xu, A multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction, Neurocomputing 419 (2021) 344–356.

[8] D. T. Araci, FinBERT: Financial sentiment analysis with pre-trained language models, arXiv preprint arXiv:1908.10063 (2019).

[9] F. Z. Xing, E. Cambria, R. E. Welsch, Natural language based financial forecasting: A survey, Artificial Intelligence Review 50 (2018) 49–73.

[10] K. Du, F. Xing, E. Cambria, Incorporating multiple knowledge sources for targeted aspect-based financial sentiment analysis, ACM Transactions on Management Information Systems (2023).

[11] W. Antweiler, M. Z. Frank, Is all that talk just noise? The information content of internet stock message boards, Journal of Finance 59 (2004) 1259–1299.

[12] P. C. Tetlock, Giving content to investor sentiment: The role of media in the stock market, Journal of Finance 62 (2007) 1139–1168.

[13] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, Journal of Computational Science 2 (2011) 1–8.

[14] C. Oh, O. R. Sheng, Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement, in: Proceedings of the International Conference on Information Systems, AIS, 2011.

[15] C. J. Hutto, E. Gilbert, VADER: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 8, 2014, pp. 216–225.

[16] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, I. Mozetič, The effects of twitter sentiment on stock price returns, PLoS ONE 10 (2015) e0138441.

[17] C. Yi-Hsuan Chen, M. Fengler, W. Härdle, Y. Liu, Textual sentiment, option characteristics, and stock return predictability, Economics Working Paper Series, University of St. Gallen (2018).

[18] I. Gurrib, F. Kamalov, Predicting bitcoin price movements using sentiment analysis: A machine learning approach, Studies in Economics and Finance 39 (2022) 347–364.

[19] X. Li, P. Wu, W. Wang, Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong, Information Processing & Management 57 (2020) 102212.

[20] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, D. C. Anastasiu, Stock price prediction using news sentiment analysis, in: 2019 IEEE 5th International Conference on Big Data Computing Service and Applications, IEEE, 2019, pp. 205–208.

[21] A. K. Davis, J. M. Piger, L. M. Sedor, Beyond the numbers: Measuring the information content of earnings press release language, Contemporary Accounting Research 29 (2012) 845–868.

[22] C.-J. Wang, M.-F. Tsai, T. Liu, C.-T. Chang, Financial sentiment analysis for risk prediction, in: Proceedings of the 6th International Joint Conference on Natural Language Processing, ACL, 2013, pp. 802–808.

[23] S. L. Scott, H. R. Varian, Predicting the present with bayesian structural time series, International Journal of Mathematical Modelling and Numerical Optimisation 5 (2014) 4–23.

[24] L. Barbaglia, S. Consoli, S. Manzan, Forecasting with economic news, Journal of Business & Economic Statistics (2022) 1–12.

[25] M. Ryan, A narrative approach to creating instruments with unstructured and voluminous text: An application to policy uncertainty, Economics Working Paper Series, University of Waikato (2020).

[26] H. Choi, H. Varian, Predicting the present with Google Trends, Economic Record 88 (2012) 2–9.

[27] L. Shang, H. Xi, J. Hua, H. Tang, J. Zhou, A lexicon enhanced collaborative network for targeted financial sentiment analysis, Information Processing & Management 60 (2023) 103187.

[28] X. Tan, Y. Cai, J. Xu, H.-F. Leung, W. Chen, Q. Li, Improving aspect-based sentiment analysis via aligning aspect embedding, Neurocomputing 383 (2020) 336–347.

[29] S. Ruder, P. Ghaffari, J. G. Breslin, A hierarchical model of reviews for aspect-based sentiment analysis, arXiv preprint arXiv:1609.02745 (2016).

[30] J. Wang, B. Xu, Y. Zu, Deep learning for aspect-based sentiment analysis, in: 2021 International Conference on Machine Learning and Intelligent Systems Engineering, IEEE, 2021, pp. 267–271.

[31] X. Wang, G. Xu, Z. Zhang, L. Jin, X. Sun, End-to-end aspect-based sentiment analysis with hierarchical multi-task learning, Neurocomputing 455 (2021) 178–188.

[32] B. Jiang, J. Hou, W. Zhou, C. Yang, S. Wang, L. Pang, Metnet: A mutual enhanced transformation network for aspect-based sentiment analysis, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 162–172.

[33] S. Yang, J. Rosenfeld, J. Makutonin, Financial aspect-based sentiment analysis using deep representations, arXiv preprint arXiv:1808.07931 (2018).

[34] S. Consoli, L. Barbaglia, S. Manzan, Fine-grained, aspect-based sentiment analysis on economic and financial lexicon, Knowledge-Based Systems 247 (2022) 108781.

[35] A. Torfi, R. Shirvani, Y. Keneshloo, N. Tavaf, E. Fox, Natural language processing advancements by deep learning: A survey, arXiv preprint arXiv:2003.01200 (2020).

[36] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, A. Balahur, Financial opinion mining and question answering, 2018. URL: `https://sites.google.com/view/fiqa/home`.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates, 2017, p. 6000–6010.

[38] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2019, pp. 2623–2631.

[39] J. Bergstra, D. Yamins, D. Cox, Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms, in: Proceedings of the 12th Python in Science Conference, volume 13, 2013, pp. 13–19.

[40] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: Proceedings of the 7th International Conference on Learning Representations, OpenReview.net, 2019.

[41] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, in: Proceeding of the 8th International Conference on Learning Representations, OpenReview.net, 2020.

[42] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, 2018, pp. 328–339.

[43] A. Salunkhe, Shubham Mhaske, Aspect based sentiment analysis on financial data using transferred learning approach using pre-trained BERT and regressor model, International Research Journal of Engineering and Technology 6 (2019) 1097–1101.

[44] H. Jangid, S. Singhal, R. R. Shah, R. Zimmermann, Aspect-based financial sentiment analysis using deep learning, in: Companion Proceedings of the Web Conference 2018, ACM, 2018, pp. 1961–1966.

[45] D. de França Costa, N. F. F. da Silva, INF-UFG at FiQA 2018 task 1: Predicting sentiments and aspects on financial tweets and news headlines, in: Companion Proceedings of the Web Conference 2018, ACM, 2018, pp. 1967–1971.

[46] G. Piao, J. G. Breslin, Financial aspect and sentiment predictions with deep neural networks, in: Companion Proceedings of the Web Conference 2018, ACM, 2018, pp. 1973–1977.