Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Analytics, Computational Intelligence and Information Management

A fused large language model for predicting startup success

Abdurahman Maarouf [a],[*], Stefan Feuerriegel [a], Nicolas Pröllochs [b]

[a] *LMU Munich & Munich Center for Machine Learning, Germany*
[b] *Justus Liebig University Giessen, Germany*

ABSTRACT

Investors are continuously seeking profitable investment opportunities in startups and, hence, for effective decision-making, need to predict a startup's probability of success. Nowadays, investors can use not only various fundamental information about a startup (e.g., the age of the startup, the number of founders, and the business sector) but also textual description of a startup's innovation and business model, which is widely available through online venture capital (VC) platforms such as Crunchbase. To support the decision-making of investors, we develop a machine learning approach with the aim of locating successful startups on VC platforms. Specifically, we develop, train, and evaluate a tailored, fused large language model to predict startup success. Thereby, we assess to what extent self-descriptions on VC platforms are predictive of startup success. Using 20,172 online profiles from Crunchbase, we find that our fused large language model can predict startup success, with textual self-descriptions being responsible for a significant part of the predictive power. Our work provides a decision support tool for investors to find profitable investment opportunities.

1. Introduction

Startups are ventures undertaken by entrepreneurs to seek, develop, and validate a business model (Katila, Chen, & Piezunka, 2012). For investors, startups represent investment opportunities with substantial financial risks yet often also with the prospect of large returns. Return on investment can easily exceed the initial investment by several orders of magnitude. As an example, the early-stage investment of Peter Thiel of 0.5 million USD into Facebook increased in value by more than 1 billion USD (CNN, 2020). However, successful investments into startups are rare. Many startups cannot establish an economic business model and eventually fail (Täuscher & Kietzmann, 2017). Given the high failure rate among startups, investors are confronted with the non-trivial decision-making task of identifying startups that will eventually be successful (Huang & Pearce, 2015; Scott, Shu, & Lubynsky, 2020).

In order to find successful startups, investors can nowadays access information about startups through online platforms for venture capital (VC). A prominent example is Crunchbase, where startups can present their venture to investors through a detailed online profile. The online profiles can include both (i) fundamental variables, which provide structured information on founders, funding, and the business sector, and (ii) textual self-description. The latter is a free text that can be used to describe the startup in verbal form. Startups can use such online profiles to inform about the venture's prospects and attract the interest of venture capitalists and other potential investors (Connelly, Certo, Ireland, & Reutzel, 2011; Parhankangas & Ehrlich, 2014).

Prior literature has explored the potential of leveraging VC platform data (e.g., from Crunchbase) to predict startup success due to their comprehensive coverage (e.g., Alamsyah & Nugroho, 2018; Arroyo, Corea, Jimenez-Diaz, & Recio-Garcia, 2019; Dellermann, Lipusch, Ebel, Popp, & Leimeister, 2017; Sharchilev et al., 2018; Weibl & Hess, 2019). However, prior studies have primarily assessed the predictive power of fundamental variables (e.g., Alamsyah & Nugroho, 2018; Arroyo et al., 2019; Dellermann et al., 2017), while mostly ignoring textual self-descriptions. Notable exceptions are Kaiser and Kuhn (2020) and Sharchilev et al. (2018), who use textual self-descriptions for prediction. However, these works rely on traditional methods with manual feature engineering. We thus contribute to the existing literature stream with a novel, fused large language model to combine textual self-descriptions with fundamental variables for predicting startup success.

In this paper, we aim to predict startup success from online profiles of VC platforms. Thereby, we not only consider fundamental information (e.g., on founders, funding, and the business sector) that are captured in traditional scorecards but we also leverage the textual self-descriptions in online profiles on VC platforms. Here, we develop a machine learning approach to predict startup success from large-scale VC platforms. Machine learning allows us to assess how well startup success can be detected for new startups and thus support the decision-making of investors regarding whether to select a startup for

funding. Specifically, we develop a tailored, *fused* machine learning approach for predicting startup success that considers both (structured) fundamental variables and (unstructured) textual self-descriptions. For this, we draw upon large language models as a recent innovation in machine learning (Devlin, Chang, Lee, & Toutanova, 2018), which we carefully adapt to our research objective. A key benefit of large language models in practice is that they are pre-trained on a large amount of public data, because of which relatively small datasets are sufficient for fine-tuning and, thus, to generate accurate predictions. We then assess the relative contribution of textual self-descriptions to making predictions of startup success.

We evaluate our machine learning approach based on our fused large language models for predicting startup success using 20,172 online profiles from Crunchbase. Crunchbase is one of the largest online VC platforms hosting online profiles from startups. We find that only fundamental variables can alone make predictions with a balanced accuracy of 72.00 %. When additionally incorporating textual self-descriptions, the balanced accuracy increases to 74.33 %. The improvement is statistically significant, implying that textual self-descriptions are effective in predicting startup success. In addition, we estimate the financial performance of our machine learning approach by translating the performance improvement to investment portfolio improvement. The investment portfolio improvement amounts to a 40.61 percentage points increase in return on investment (ROI) when incorporating textual self-descriptions, highlighting the practical implications of our machine learning approach. We then evaluate the prediction performance across various events indicating startup success (i. e., initial public offering, acquisition, and external funding). We further provide an extensive series of sensitivity analyses in which we compare the prediction performance across business sectors, startup age, and additional machine learning baselines, thereby confirming the robustness of our findings.

Our work contributes to business analytics in several ways. First, we provide empirical evidence on the operational value of online VC platforms for better investment decision-making. Thereby, we extend upon extensive research which has studied the benefits of online platforms for users while we focus on investors. Second, we contribute to a growing stream of machine learning in business analytics (e.g., Bastani, Zhang, & Zhang, 2022; Choi, Wallace, & Wang, 2018; Cohen, 2018; Mišić & Perakis, 2020). Here, we demonstrate an impactful application of machine learning in VC decision-making. Third, we show the operational value of large language models for research and practice. However, as we detail later, a naïve application of large language models would miss significant predictive power. Instead, our task requires a non-trivial adaptation through a *fused* large language model to our decision problem in order to make combined predictions from both fundamental information and texts. Fourth, we provide a flexible tool for investors to automate their screening process in VC decision-making.

The rest of this paper is structured as follows. Section 2 provides a background on venture capital and analytics for decision-making. In Section 3, we develop our machine learning approach in the form of a tailored large language model. Section 4 presents our dataset with online profiles from Crunchbase, based on which we study the predictive power of textual self-descriptions (Section 5). We then discuss implications for both business analytics practice and research (Section 6), while Section 7 concludes.

2. Related work

2.1. Venture capital

Startups are new entrepreneurial ventures founded to develop and validate a business model (Katila et al., 2012). In practice, startups typically take an innovative idea and then build a scalable business model around it, with the intention of turning the startup into a high-growth, profitable company (Forbes, 2019). This process is largely dependent on external funding in order to cover costs for technology development, entering markets, or other upfront investments. Hence, events in which startups receive funding are commonly used in the literature to determine success (Arroyo et al., 2019; Hegde & Tumlinson, 2014; Nanda & Rhodes-Kropf, 2013). Examples of such events are initial public offerings (e. g., Airbnb, which went public in December 2020), acquisitions (e. g., Slack and DeepMind, which were acquired by Salesforce and Google, respectively), and external funding (e. g., SpaceX, which had several funding rounds after its series A funding in 2002). To capture startup success, prior literature has often studied either individual events such as initial public offerings (e. g. Chang, 2004; Mann & Sager, 2007) or a combination of events (e. g. Arroyo et al., 2019; Hegde & Tumlinson, 2014; Nanda & Rhodes-Kropf, 2013).

Startups often represent lucrative investment opportunities with the prospect of large returns. As of 2023, more than 180 startups have turned into "unicorns", that is, reached valuations of over USD 1 billion in less than five years (Fleximize, 2023). Investing in such unicorns in an early stage can create a return multiple times larger than the initial investment. However, investments in startups are known to be of high risk. Startups that eventually fail leave the investor with little or no return. Hence, identifying successful startups at an early stage is difficult (Aggarwal & Singh, 2013; Huang & Pearce, 2015; Scott et al., 2020).

Predicting which startups will turn out to be successful is inherently challenging, as startups represent new ventures for which little to no information on past performance is available. Thus, many investors make such predictions based on their gut feeling (Huang & Pearce, 2015). However, according to prior literature, there are several determinants that characterize successful startups. These can be loosely divided into characteristics regarding the business model, the founders, and funding. (1) The business model explains – to some degree – the survival of ventures (Böhm, Weking, Fortunat, Müller, Welpe, & Krcmar, 2017; Weking, Böttcher, Hermes, & Hein, 2019). In this regard, the business sector is also associated with startup success (van Gelderen, Thurik, & Bosma, 2005; Holmes, Hunt, & Stone, 2010). (2) Founders decide upon how a business is run and thus founder characteristics are important success factors (e. g., Littunen & Niittykangas, 2010; Lussier, 1995). For instance, startups are more likely to be on a path towards growth when their founders have attended higher education (van Gelderen et al., 2005). (3) Funding is often a prerequisite to stimulate growth (e. g., Butler, Garg, & Stephens, 2020; Conti & Graham, 2020; Lussier, 1995). Hence, startup success is also associated with previous funding rounds (Baum & Silverman, 2004). In this regard, it is further beneficial for startups to have backing from a known venture capitalist (Conti & Graham, 2020; Nahata, 2008). Hence, to avoid relying on gut feeling or subjective bias when processing information about startups, machine learning presents a scalable, data-driven approach to predict startup success.

2.2. Predicting startup success

Prior works have developed data-driven approaches for predicting startup success. For instance, predictions can be made based on data from questionnaires, namely via so-called scorecards (Böhm et al., 2017; Yankov, Ruskov, & Haralampiev, 2014). One study also draws upon data that are extracted from business plan competitions (McKenzie & Sansone, 2017). Yet, both questionnaires and business plan competitions involve data from manual reporting, which is often not available in VC markets. These data sources also tend to have limited coverage, and thus their usefulness in the daily decision-making of investors is limited. A different stream of literature predicts acquisitions as a specific event in startup lifecycles using proprietary databases (e.g., COMPUSTAT Ragothaman, Naik, & Ramakrishnan, 2003, SDC

Platinum Wei, Jiang, & Yang, 2008). However, such databases are typically restricted to specific events (i. e., acquisitions) and, on top of that, have limited coverage as they provide only few variables (e.g., about founders and funding) but not textual descriptions. In contrast to that, textual descriptions about startups and their business model, innovation, or market structure may provide significant predictive power regarding which ventures will eventually be successful.

Recently, the possibility of using online data from VC platforms to predict startup success was explored (Alamsyah & Nugroho, 2018; Arroyo et al., 2019; Dellermann et al., 2017; Sharchilev et al., 2018). Predicting startup success from VC platforms has a clear advantage in practice: online platforms for VC typically exhibit comprehensive coverage of startups and thus provide "big" data (Weibl & Hess, 2019). This is beneficial, as large-scale datasets are generally a prerequisite for making accurate inferences using machine learning. Studies predicting startup success based on questionnaires have often relied on samples with less than 200 observations (i. e., 200 different startups), because of which the prediction models can *not* generalize well across startups and thus lack predictive power (Böhm et al., 2017; Yankov et al., 2014). In contrast, online platforms for VC, such as Crunchbase, provide online profiles of more than 20,000 startups in the U. S alone.

Based on data from VC platforms, a variety of research questions have been studied. Arroyo et al. (2019) evaluate the predictive ability of fundamental information at Crunchbase, but textual self-reports as predictors are ignored. In Dellermann et al. (2017), a hybrid machine learning approach is designed in which both fundamental information and judgment scores from crowds are combined, but again textual self-reports as predictors are again ignored. Sharchilev et al. (2018) use textual self-reports for prediction but rely on traditional, bag-of-words representations and not large language models. Kaiser and Kuhn (2020) also make predictions from self-reports but rely on a dictionary-based approach that requires manual feature engineering. Hence, the ability of large language models together with textual self-descriptions from VC platforms has yet to be explored and presents our contribution.

2.3. Machine learning in business analytics

Machine learning can support managerial decision-making by predicting uncertain operational outcomes (Choi et al., 2018; Cohen, 2018; Feuerriegel, Shrestha, von Krogh, & Zhang, 2022; Kraus, Feuerriegel, & Oztekin, 2020). The adoption of machine learning in business analytics has been greatly fueled by the increasing availability of data and recent methodological advances (Bastani et al., 2022; Mišić & Perakis, 2020). Promising examples include credit scoring (Kriebel & Stitz, 2022; Lessmann, Baesens, Seow, & Thomas, 2015; Maldonado, Pérez, & Bravo, 2017; Verbraken, Bravo, Weber, & Baesens, 2014), financial risk assessment (Kim et al., 2020; Kozodoi, Jacob, & Lessmann, 2022), business failure prediction (Borchert, Coussement, De Caigny, & De Weerdt, 2023; De Bock, Coussement, & Lessmann, 2020; Naumzik, Feuerriegel, & Weinmann, 2022; Stevenson, Mues, & Bravo, 2021), throughput prediction (Senoner, Kratzwald, Kuzmanovic, Netland, & Feuerriegel, 2023), customer analytics (De Caigny, Coussement, & De Bock, 2018; Ozyurt, Hatt, Zhang, & Feuerriegel, 2022), recommendation systems (Geuens, Coussement, & De Bock, 2018), and public sector operations (Jakubik & Feuerriegel, 2022; Kadar, Maculan, & Feuerriegel, 2019). However, the aforementioned works build upon structured data and not text.

Business analytics has also increasingly embraced machine learning that can make inferences from textual content (e.g., Borchert et al., 2023; Cui, Gallino, Moreno, & Zhang, 2018; Feuerriegel & Gordon, 2019; Feuerriegel, Hartmann, Janiesch, & Zschech, 2024; Haupt, Bender, Fabian, & Lessmann, 2018; Kraus & Feuerriegel, 2017; Kriebel & Stitz, 2022; Lau, Zhang, & Xu, 2018; Stevenson et al., 2021; Toetzke, Banholzer, & Feuerriegel, 2022). As such, business analytics can mine user-generated content, e.g., from social media, in an automated and scalable manner (Cui et al., 2018). For example, Cui et al. (2018)

enrich historical sales data with social media as a measure of customer perception towards products and evaluate how that combined data source is better in predicting future sales. However, existing methods in business analytics oftentimes build upon bag-of-words approaches where an unordered set of words is used as input (e.g., Cui et al., 2018; Feuerriegel & Gordon, 2019; Lau et al., 2018) and where, as a result, the relationship, order, and hierarchical structure among words is lost. Hence, existing methods merely operate on word frequencies and not on semantic meaning. A potential remedy is given by large language models that model the ordered sequence of words and thus capture the semantics of running text; however, the operational value of large language models has so far been largely unclear. Moreover, we are not aware of previous work that uses large language models for startup prediction to support investment decisions.

3. Empirical model

In this section, we first formulate our research question of whether textual self-descriptions from VC platforms predict startup success. To answer this, we then describe our machine learning approach based on a tailored, fused large language model.

3.1. Research question

In this study, we build a machine learning approach where we leverage information provided by startups on VC platforms in order to predict startup success. Information on online VC platforms such as Crunchbase can loosely be grouped into two categories (which may potentially complement each other). (1) VC platforms provide structured information on a startup's fundamentals. Examples of such fundamental variables are the age of the startup, the number of founders, or information about past funding success. Fundamental variables are typically entered on VC platforms in a structured format and thus with little degree of customization. (2) Startups can additionally provide a textual self-description. The textual self-description can be used to describe the business model, a startup's innovation, or the market structure. Textual self-descriptions have become mandatory on VC platforms such as Crunchbase but the actual content is at the startup's discretion.

In this study, we examine whether large language models can be successfully leveraged by investors to predict startup success from textual self-descriptions on VC. There are several factors that lead us to expect that textual self-descriptions are predictive. In particular, startups can use the textual self-description on VC platforms to present information on a startup's business model, innovation, or market structure. An example is "BetterTrainers has a new type of business model that protects all sessions booked through the site with premium insurance coverage" where a business model is explained, or "FaceTec's patented, industry-leading 3D Face Authentication software anchors digital identity with 3D FaceMaps" where a startup details how to make use of certain technologies. Besides the actual information captured in the text, latent factors such as the tone of the text (e.g., a positive sentiment) may also implicitly signal success.

As seen by the previous examples, traditional approaches from machine learning for making predictions from online descriptions (e.g., bag-of-words) will likely struggle with the complexity of the underlying task since traditional approaches only rely upon word frequencies and do not provide a principled approach to infer semantic meanings. To this end, the previous examples motivate the use of large language models in our study as a principled, data-driven approach to capture semantic meanings in text and thus to predict startup success.
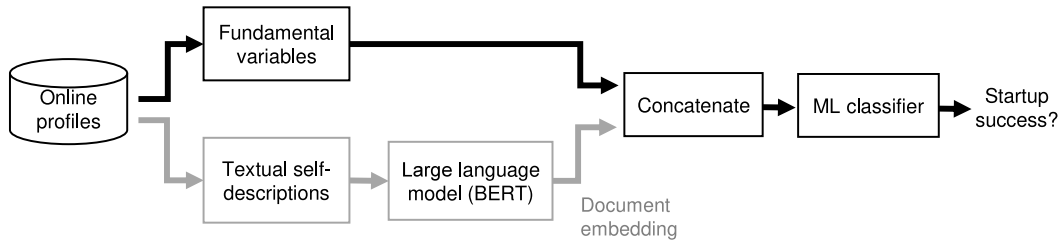
Fig. 1. Our machine learning approach based on a tailored, fused large language model for predicting startup success.

3.2. Fused machine learning approach

In the following, we present our *fused* machine learning approach in order to predict startup success. Let $i = 1, \ldots, n$ denote the startups. Specifically, we develop a tailored, fused large language model as shown in Fig. 1. In our machine learning approach, both sets of variables – i.e., fundamental variables (FV) and textual self-descriptions (TSD) – are taken into account but in different ways. (1) The fundamental variables come in a structured format $x_i^{\text{FV}} \in \mathbb{R}^{m_{\text{FV}}}$ and are thus directly passed on to the final machine learning classifier. (2) The textual self-descriptions are first mapped onto document embeddings $x_i^{\text{TSD}} \in \mathbb{R}^{m_{\text{TSD}}}$ and then passed on to the final ML classifier. Let us denote the final ML classifier by $\phi_\theta : \mathbb{R}^{m_{\text{FV}} + m_{\text{TSD}}} \to \{0, 1\}$ with some parameters θ. Here, the output $y_i \in \{0, 1\}$ indicates whether a startup $i = 1, \ldots, n$ will be successful ($y_i = 1$) or not ($y_i = 0$). Crucially, a custom architecture for our large language model is necessary in order to fuse both fundamental variables x_i^{FV} and document embeddings x_i^{TSD} to make predictions. For comparison, we later evaluate a naïve large language model without the "fused" structure which uses only x_i^{TSD} for prediction.

In our machine learning approach, we take the textual self-description of the startup and use a large language model (i. e., BERT; see Devlin et al., 2018) as an embedding generator to map text onto a document embedding. The document embedding is then concatenated with the fundamental variables and the resulting concatenated vector is then used as input to the classifier. Large language models represent state-of-the-art techniques for modeling natural language in machine learning (Jurafsky & Martin, 2020). A prominent example is BERT, which has been found effective in capturing complex dependencies such as semantics in textual content (Devlin et al., 2018). In the following, we detail how we fuse data in our large language model.

3.2.1. Large language model (BERT) as embedding generator

Large language models, often also called transformers, are large-scale deep neural networks that are carefully designed to process running text (Jurafsky & Martin, 2020). The practical benefit of large language models is that they leverage the strength of large-scale deep neural networks and are thus able to capture context, semantics, structure, and meaning (Jurafsky & Martin, 2020).

A prominent large language model is BERT (Devlin et al., 2018). BERT was developed by Google AI and stands for bidirectional encoder representations from transformer. BERT has been successful in solving various machine learning tasks for natural language. In particular, BERT has been shown to be superior to alternative document representations such as bag-of-words. Methodologically,

Language models such as BERT map running text onto a new representation called embedding (Devlin et al., 2018). Formally, each textual input is first transformed into a sequence of tokens $[[\text{CLS}], w_1, w_2 \ldots]$ based on the predefined vocabulary of BERT, with [CLS] being used at the beginning of each sequence. Hence, for each textual input, BERT receives a sequence of individual tokens as input where the tokens are represented by vectors $[\text{CLS}], w_1, w_2, \ldots \in \mathbb{R}^w$. The vectors are not "one-hot-encoded" as traditionally done in simpler models. Instead, BERT uses an embedding layer to convert the sequence of tokens into dense vector representations $e_{[\text{CLS}]}, e_1, e_2, \ldots \in \mathbb{R}^e$ that are lower-dimensional (i.e., the dimensionality e is much smaller than the dimensionality of a typical one-hot encoding, which is computationally more desirable). Next, the token embeddings are fed into a transformer encoder. A transformer encoder is a neural network designed for sequential data that processes the entire input sequence $[e_{[\text{CLS}]}, e_1, e_2, \ldots]$ simultaneously, rather than sequentially. It relies on two key mechanisms: (a) positional encodings, which add information about the position of each token to retain the order; and (b) an attention mechanism, which allows the model to weigh the importance of different tokens dynamically. Thereby, a transformer encoder employs a complex, non-linear process to determine how tokens influence one another. The output of the transformer encoder consists of transformed vectors (embeddings) $[o_{[\text{CLS}]}, o_1, o_2, \ldots]$, which can then be used for various tasks. Specifically, the embedding for the [CLS] token (i.e., $o_{[\text{CLS}]}$) can be used for classification tasks as it aggregates the meaning of the entire input sequence.

During training, BERT utilizes a technique called masked language modeling, where some of the input tokens are randomly masked (i.e., omitted) for self-supervised learning. The objective of BERT during training is to correctly predict these masked tokens. Thereby, BERT updates its internal weights and learns a deep understanding of language context and relationships between words. Due to self-supervised learning, large-scale textual databases (e.g., Wikipedia) can be used for training but without the need for explicitly annotated labels. A schematic visualization is in Fig. 2.

Our implementation is as follows. We use the so-called basic, uncased version of BERT (Devlin et al., 2018), comprised of 12 layers with ∼110 million trainable parameters. It generates embeddings $o_{[\text{CLS}]}, o_1, o_2, \ldots \in \mathbb{R}^n$ of dimension $n = 768$. BERT is shipped as a pre-trained network where parameters have already been learned from open-source content. Before applying BERT, all text is lowercased and tokenized using the WordPiece algorithm, which maps the text onto subwords or unigrams from the WordPiece vocabulary. Afterward, the text is passed through the pre-trained BERT network. The embedding of the [CLS] token (i.e., $o_{[\text{CLS}]}$) is then used as the document embedding x^{TSD}, representing the textual self-description for the downstream classification. Hence, our document embedding x^{TSD} is of dimension $m_{\text{TSD}} = 786$.

3.2.2. Baseline text representations

We compare our machine learning approach based on a tailored, fused large language model against three traditional text representations. All of the baselines are again concatenated to the fundamental variables and are then fed into a final machine learning classifier. The final machine learning classifier is again subject to rigorous hyperparameter tuning (see later for details) for fair comparison. Therefore, all performance gains from our approach must be attributed to that large language models are better at handling textual content.
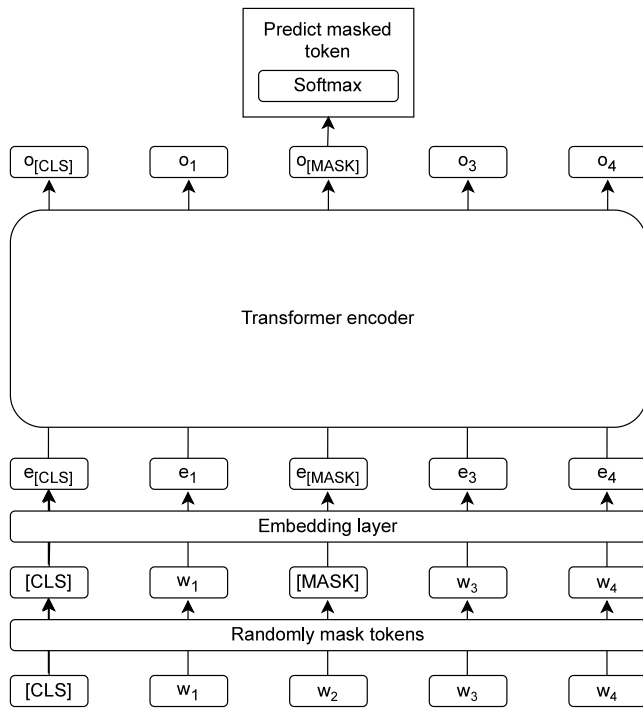
Fig. 2. Schematic overview of a large language model (here: BERT).

- **Manual feature extraction:** The first text-based baseline is based on manual feature extraction. Specifically, we manually craft features that capture textual information (e.g., the length, the mean word length, and the number of geographic references). We follow prior literature and extract the same features as in Kaiser and Kuhn (2020). This results in a text representation of dimension 10. We refer readers to Kaiser and Kuhn (2020) for a full list of the features.

- **Bag-of-words:** We compare our machine learning approach against the traditional approach of a bag-of-words baseline. We refer readers to Jurafsky and Martin (2020) for an introduction. We implement bag-of-words as follows. We first tokenize the words of the textual self-description to unigrams, remove stop words, lemmatize, and apply a tf-idf weighting. Furthermore, we remove words with more than 95% sparsity. The bag-of-words baseline results in a 98-dimensional text representation.

- **GloVe:** GloVe (Pennington, Socher, & Manning, 2014) transforms words into vectors (so-called word embeddings) based on their co-occurrence in a text corpus. Thereby, the vectors capture semantic relationships, offering a rich set of features for text analysis. We use the GloVe model pre-trained on Wikipedia (i.e., `glove-wiki-gigaword-50`) to extract the 50-dimensional word embeddings. We average the individual word embeddings to get the final text representation used for the downstream classification.

3.2.3. Final machine learning classifier

The final machine learning classifier $\phi_\theta(\cdot)$ with parameters θ is responsible for the "fused" approach and, for this, receives the concatenated vector of (1) fundamental variables and (2) document embeddings. The output is then the predicted probability of startup success. We thus optimize

$$\theta^* = \arg\min_{\theta} \mathbb{E}[\mathcal{L}(\phi_\theta([x^{\text{FV}}, x^{\text{TSV}}]), y)], \tag{1}$$

where $\mathcal{L}$ is a convex loss (e.g., mean squared error) and where $[\cdot, \cdot]$ is the concatenation operator.

We experiment with different classifiers that are designed to handle both linear and non-linear relationships in the data. Specifically, we make use of the following classifiers:

- **Logistic regression:** The logistic regression is a simple linear model used for binary classification. It models the probability of a binary outcome using the logit function to map predictions to probabilities. The logistic model expresses the log-odds of the outcome variable as a linear combination of the independent variables, formalized as $\log\left(\frac{p}{1-p}\right) = \theta^T x$, where p is the probability of the outcome of interest.

- **Elastic net:** The elastic net extends the logistic regression in which overfitting is prevented through regularization (Zou & Hastie, 2005). Specifically, regularization is given by a combination of both an L1- and an L2-norm penalty (analogous to lasso and ridge methods, respectively). This thus shrinks some coefficients closer to zero, and, as a result, the classifier generalizes better to out-of-sample observations. Formally, let $\phi_\theta(x) = \theta^T x$. Then the regularized loss $\mathcal{L}_{\text{reg}}$ is formalized as $\mathcal{L}_{\text{reg}}(x, y) = \mathcal{L}(\phi_\theta(x), y) + \lambda\left(\frac{1-\alpha}{2}\|\theta\|_2^2 + \alpha\|\theta\|_1\right)$ with hyperparameters α and λ. The elastic net is especially beneficial in tasks where predictors are subject to linear dependence (Hastie, Tibshirani, & Friedman, 2009). For reasons of completeness, we also experimented with lasso and ridge methods (Hastie et al., 2009), but with qualitatively similar results (and thus omitted the results for brevity).

- **Random forest:** The random forest is an ensemble learning classifier where predictions are made from a multitude of decision trees (Ho, 1995). Each decision tree is fit to a random subset of the data, while the final prediction is then made by taking the majority vote over the individual decision trees. As a result, the classifier is less prone to overfitting, has a better prediction performance than a single decision tree, and is effective in handling non-linear relationships (Hastie et al., 2009).

- **Neural net:** The neural network is a flexible model for classification by using layers of nodes that transform the input through non-linear activation functions. The output layer uses a sigmoid to produce class probabilities. The loss is regularized by a combination of both the L1- and L2-norm penalty to prevent overfitting. Neural networks excel due to their flexibility in handling non-linear relationships.

3.3. Performance metrics

To evaluate the performance of machine learning in predicting startup success, we report different performance metrics: balanced accuracy, precision, recall, F_1-score, area under the curve from the precision–recall curve (AUCPR), and area under the curve from the receiver operating characteristics (AUROC). However, due to its inherent benefit of considering the complete distribution of discrimination thresholds (Hastie et al., 2009), we primarily focus on the AUROC. We remind that we follow common practice in machine learning and evaluate the performance on out-of-sample observations, that is, startups that have not been part of the training set but from the test set so that they are thus unseen to the machine learning classifiers.

Furthermore, we calculate the return on investment (ROI) for the machine learning-selected portfolios. Let TP denote the number of true positives (i.e., cases where startup success was predicted correctly) and FP the number of false positives (i.e., cases where the model predicted success despite that the startup was actually not successful). We then calculate the net investment gain for correctly predicted successful startups by taking the sum of the final investment values FIV_{TP} (i.e., the startup valuations after a success event) minus the sum of the total cost of investment (IC). Note that, since data on startup valuations and costs of investment is not always publicly available, we approximate these variables using constants determined based on historical mean

Table 1
Tuning grid for hyperparameter tuning.

Classifier	Hyperparameter	Tuning range
Logistic regression	—	
Elastic net	L1 ratio	{0.1, 0.2, 0.3, 0.5}
	Tolerance for stopping	{0.00001, 0.0001, 0.001, 0.01}
	Regularization parameter	{0.1, 0.25, 0.5, 1, 2, 4, 8}
Random forest	% of randomly selected predictors	{0.4, 0.6, 0.8}
	Splitting rule	{gini, entropy}
	Minimal node size	{5, 8, 10}
	Minimal split size	{8, 10}
Neural network	# Hidden units	$input\_dim * \{0.25, 0.5, 0.75, 1, 1.5, 2\}$
	# Hidden layers	2, 3, 4
	Dropout rate	{0, 0.2}
	Batch size	{128, 256, 512}
	Optimizer	AdamW (Loshchilov & Hutter, 2018)
	Learning rate	{0.001, 0.0001, 0.00001}
	Max Epochs	500
	Learning rate decay	0
	Activation function	ReLU
	Early stopping patience	3

Note: $input\_dim$ is the number of input features and therefore depends on whether the fundamental variables, the textual self-description, or a combination of both is used for training.

values for startups listed on Crunchbase.[1] For companies that were non-successful, we conservatively assign a final investment value of zero (FIV_{FP}). The ROI for the portfolio is then calculated by taking the net investment gain divided by the total cost of investment. Formally,

$$ROI = \frac{TP \times FIV_{TP} + FP \times FIV_{FP} - (TP + FP) \times IC}{(TP + FP) \times IC} \times 100, \quad (2)$$

where the total investment costs (IC) comprise (i) the investor's investment into equity of the startup; and (ii) we consider 10% of the last valuation as additional screening and monitoring costs for the investor.

3.4. Implementation details

Our implementation follows best practice in machine learning (Hastie et al., 2009). For this, we split the data into a training set and a test set. The former is used for training the model; the latter is used to evaluate the out-of-sample performance. In our work, we randomly assign 80% of the data samples to the training set and 20% to the test set. Due to class imbalances, common procedures in machine learning are followed; that is, we apply a stratified split (Goodfellow, Bengio, & Courville, 2016), so that both sets have the same ratio of successful vs. non-successful startups. To ensure robustness in our evaluation, we repeat the random split five times and report the mean and standard deviation of the performance metrics on the test set across the five iterations. This allows us to quantify how well machine learning can predict success for ventures that were not seen during training.

Hyperparameter tuning is conducted using 10-fold cross-validation. Specifically, hyperparameters are tuned via randomized grid search (20 iterations), using the tuning grid in Table 1. The best hyperparameters are selected based on the cross-validated AUROC score. Note that the hyperparameter tuning is done separately for the different input variables, that is, for when training our machine learning approach using fundamental variables (FV), textual self-descriptions (TSD), or a combination of both (FV + TSD).

4. Empirical setting

4.1. Online profiles on crunchbase

Our evaluations are based on data from Crunchbase.[2] Crunchbase is a leading online VC platform that connects startups and investors. For this, Crunchbase allows startups to create online profiles where they can present information on their business, founders, and funding. Edits can be made by verified employees to ensure that correct information is entered.

We collected online profiles (i. e., both fundamental variables and textual self-descriptions) from all US-based startups that were listed on Crunchbase. Furthermore, we excluded startups that went public and that have already received series C funding (or a later funding round). The latter is important as our objective is to make predictions for companies that fall under the definition of a startup.

4.2. Definition of startup success

In our study, we predict startup success with regard to different events that are conventionally used as indicators of success (cf. Arroyo et al., 2019; Hegde & Tumlinson, 2014; Nanda & Rhodes-Kropf, 2013), namely, whether startups had an initial public offering, have been acquired, or secured external funding. If any of these events occurred, we treat the startup as "successful". Otherwise, a startup is treated as "non-successful". If not stated otherwise, these labels are used to evaluate our machine learning approach. As part of our sensitivity analyses, we later continue to compare how the prediction performance varies across these events – i. e., initial public offerings, funding, and acquisitions.

4.3. Time-aware prediction and evaluation framework

We implemented a *time-aware* approach that is common in time-series forecasting (Hastie et al., 2009). Recall that we aim to evaluate whether we can predict if startups will become successful in the future. Consequently, we processed our data as follows. We restricted our analysis to startups that were founded between 2013 and 2015, based on which we predicted their future development until the end of 2020. We obtained raw access to the Crunchbase database with historical data. This allowed us to collect information from online profiles that were

---

[1] In our Crunchbase dataset (see Section 4), the valuation of a startup after a success event (i. e., initial public offerings, funding, acquisitions) is, on average, $184.47 million. The pre-success valuation (i. e., the last valuation in previous funding rounds) is, on average, $12.19 million. Hence, startups have, on average, a 15.13 times higher valuation if they become successful.

[2] http://www.crunchbase.com

Table 2
Variable descriptions.

Variable	Description
	OUTCOME VARIABLE
Success	True (= 1) if a startup had an initial public offering, received funding, or has been acquired. False (= 0) otherwise.
	PREDICTORS
Fundamental variables (FV)	
Age	Time since the startup has been founded (in months)
Has email	Whether the startup has added an email address (= 1 if true, otherwise 0)
Has phone	Whether the startup has added a phone number (= 1 if true, otherwise 0)
Has Facebook	Whether the startup has added a link to Facebook (= 1 if true, otherwise 0)
Has Twitter	Whether the startup has added a link to Twitter (= 1 if true, otherwise 0)
Has LinkedIn	Whether the startup has added a link to LinkedIn (= 1 if true, otherwise 0)
Founders count	Number of founders of the startup
Founders country count	Number of unique countries the founders are from
Founders male count	Number of male founders
Founders female count	Number of female founders
Founders degree count total	Total number of university degrees of the founders
Founders degree count maximum	Number of degrees for most educated founder
Founders degree count mean	Average number of degrees per founder
Number of investment rounds	Number of investment rounds
Raised funding	Total raised funding (in million USD)
Last round investment type	Investment type (seed, series A, etc.) of the last funding round
Last round raised funding	Raised funding (in million USD) in the last funding round
Last round post money evaluation	Valuation (in million USD) of the startup after the last funding round
Last round time lag	Time since last funding round (in months)
Investor count	Overall number of investors that invested in the startup
Last round investor count	Number of investors in the last funding round only
Known investor count	Overall number of investors with a profile on Crunchbase
Last round known investor count	Number of investors in the last funding round with a profile on Crunchbase
Industries	Fine-grained industries in which the startup operates (according to the Crunchbase coding scheme; e.g., "machine learning", "machinery manufacturing")
Textual self-description (TSD)	
Document embedding	Textual self-description encoded via large language model (BERT)

available in 2015. In particular, we discarded information that was added or updated later, so that we only considered data as presented on Crunchbase at the end of 2015.

We then predict whether an event indicating startup success has occurred during the years 2016 through 2020, that is, we make forecasts whether startups were successful over a time horizon of five years. The forecast horizon is set analogous to earlier statistics reporting upon a high failure rate among startups in their early stage (U.S. Bureau of Labor Statistics, 2016), so that a 5-year-ahead forecast horizon should be sufficient to distinguish successful from non-successful startups. Our choice of events representing startup success is listed in the previous section.

4.4. Variable descriptions

Our fused machine learning approach makes use of an extensive set of variables from Crunchbase (see Table 2). The outcome variable (i. e., the variable to predict) is binary, denoting whether a startup was successful (= 1; otherwise = 0).

The predictors (i. e., the variables that are fed into our machine learning classifiers) consist of the following: (structured) fundamental variables and (unstructured) textual self-descriptions. (1) The fundamental variables (FV) describe different characteristics of startups such as their age or the industries in which they operate (see Table 2). Note that we use the industries as reported on Crunchbase, which is based on a highly granular scheme (e.g., an Internet-of-Things company may be assigned simultaneously to "artificial intelligence", "industrial automation", etc.). Social media activity has been found to be related to startup success (Jin, Wu, & Hitt, 2017), and, analogously, we include information about whether startups are on social media (e. g., whether they have a Twitter/X or LinkedIn profile). Furthermore, we collect information about the characteristics of the founders (e. g., the number of university degrees). We follow previous literature (Conti & Graham,

2020) by controlling for the presence of known investors that have a profile on Crunchbase themselves. We also include information on previous funding rounds but, since we use a historical view on Crunchbase data, we only access information up to our time point when making the predictions so that there is *no* lookahead bias (i.e., we discard funding rounds that occur during the forecast horizon to ensure a time-aware evaluation framework).[3] (2) The latter, i. e., textual self-descriptions (TSD), are encoded via the large language model (BERT). This yields document embeddings, which are then used as input to the machine learning classifier.

4.5. Descriptive statistics

The above filtering yields a final dataset with 20 172 startups. Descriptive statistics on startups for our dataset are as follows (see Table 3). Out of all startups, 7252 (i. e., 35.94%) startups have been labeled as successful, whereas 12 920 (i. e., 64.06%) have been labeled as being non-successful. Frequent events indicating success are founding rounds (i. e., 32.45% of all startups), followed by acquisitions (3.10%) and initial public offerings (0.40%). For startups in our dataset, the average age is 18 months. Startups tend to be more successful if they provide a link to their social media profiles. In general, startups are more frequently founded by males (i. e., 1.58 male founders per startup)

---

[3] We also considered a less sparse encoding of business sectors (rather than fine-grained industries as in the Crunchbase coding scheme) but we discarded this. The reason is seen in our later analysis, where there is little variability across business sectors and, thus, sector information has only little predictive power. Furthermore, we also considered additional information about founders (e.g., their number of current and past jobs) and prominence (e.g., site visitors, growth in site visitors, number of media articles) but found that these are too sparse to make a meaningful addition to our predictions.

Table 3
Descriptive statistics.

Variable	Overall		Non-successful		Successful	
	Mean	SD	Mean	SD	Mean	SD
OUTCOME VARIABLE						
Success	0.36	0.48	0.00	0.00	1.00	0.00
FUNDAMENTAL VARIABLES (FV)						
Age (in months)	18.14	10.01	19.54	9.85	15.66	9.81
Has email	0.77	0.42	0.75	0.43	0.82	0.38
Has phone	0.58	0.49	0.58	0.49	0.58	0.49
Has Facebook	0.73	0.45	0.73	0.44	0.72	0.45
Has Twitter	0.77	0.42	0.75	0.43	0.79	0.41
Has LinkedIn	0.70	0.46	0.62	0.49	0.84	0.36
Founders count	1.83	0.97	1.69	0.89	1.98	1.03
Founders different country count	1.20	0.42	1.17	0.39	1.23	0.45
Founders male count	1.58	1.03	1.43	0.94	1.74	1.08
Founders female count	0.25	0.52	0.26	0.52	0.24	0.51
Founders degree count total	1.16	1.46	0.88	1.22	1.46	1.62
Founders degree count maximum	0.84	0.92	0.69	0.88	1.00	0.94
Founders degree count mean	0.78	0.84	0.66	0.83	0.92	0.83
Number of investment rounds	1.36	0.73	1.25	0.61	1.51	0.83
Raised funding (in million USD)	3.16	21.45	1.83	21.58	4.81	21.17
Last round raised funding (in million USD)	2.59	22.23	1.73	22.92	3.85	21.12
Last round post money evaluation (in million USD)	12.19	48.39	8.48	36.06	16.84	60.19
Last round time lag (in months)	11.59	8.55	14.14	8.80	8.40	7.03
Investor count	2.07	3.82	1.11	2.55	3.28	4.70
Last round investor count	1.42	2.65	0.79	1.88	2.34	3.27
Known investor count	1.16	0.68	1.06	0.38	1.30	0.91
Last round known investor count	1.08	0.46	1.02	0.25	1.16	0.62
TEXTUAL SELF-DESCRIPTIONS (TSD)						
TSD length in chars (only for descriptive purpose)	665.02	429.26	694.04	462.13	613.32	357.64

SD = standard deviation   *N* = 20,172 startups

Table 4
Exemplary textual self-descriptions for each class.

Textual self-description	Outcome
"Fluc (which is Miles & Company Services now) is building the world's first social marketplace for consumers to search, discover, and purchase freshly-made food. Whether people desire a cup of coffee or a freshly cooked pacific trout, Fluc powers the connection between consumers and local food merchants. Fluc wraps complex logistics into a simple and affordable consumer experience, enabling anyone to access thousands of food items from the palm of their hand."	non-successful
"Lemonade is a licensed insurance carrier that offers homeowners and renters insurance powered by artificial intelligence and behavioral economics. By replacing brokers and bureaucracy with bots and machine learning, Lemonade promises zero paperwork and instant everything. And as a Certified B-Corp, where underwriting profits go to nonprofits, Lemonade is remaking insurance as a social good, rather than a necessary evil."	successful

than by females (i. e., 0.25 female founders per startup). Successful startups have, on average, more founders (mean: 1.98) than non-successful ones (mean: 1.69). Furthermore, founders with university degrees often have more successful startups. On average, startups have previously raised funding totaling to USD 3.156 million from 2.07 investors. Unsurprisingly, startups that are eventually labeled as successful have received more funding (mean: USD 4.81 million) and are backed by more investors (mean: 3.28). On average, successful startups provide a shorter textual self-description (mean: 613.32 characters) than non-successful ones (mean: 694.04 characters). Table 4 lists two example textual self-descriptions, one for each class.

Startups listed on Crunchbase operate in a variety of business sectors (see Table 5). The majority of startups in our data operate in the area of INFORMATION TECHNOLOGY and COMMUNICATION SERVICES. In contrast, startups in the ENERGY, UTILITIES, and MATERIALS sectors are less common. Note that startups can be assigned to multiple business sectors. Across the business sectors, we also see variation in the success rate of startups. For instance, startups in some sectors such as UTILITIES have a high success rate (50.79%), while the success rate for COMMUNICATION SERVICES amounts to only 32.71%.

Table 5
Relative frequencies and success rates of startups across different business sectors.

Business sector	Relative freq. (in %)	Success rate (in %)
INFORMATION TECHNOLOGY	54.32	40.61
COMMUNICATION SERVICES	43.23	32.71
INDUSTRIALS	34.72	41.53
CONSUMER DISCRETIONARY	34.14	32.95
HEALTH CARE	16.21	52.06
CONSUMER STAPLES	15.86	42.06
FINANCIALS	10.01	41.14
REAL ESTATE	5.99	35.84
UTILITIES	2.21	50.79
ENERGY	2.16	44.14
MATERIALS	1.52	43.97

Note: Business sectors are categorized according to the Global Industry Classification Standard (GICS). Startups can belong to multiple business sectors.

5. Empirical findings

5.1. Comparison of our large language model against the baselines

We now evaluate the performance of our fused large language model in predicting startup success (see Table 6). We use the neural net as the best-performing final machine learning classifier within our fused large language model for this evaluation. For a detailed comparison across different final machine learning classifiers, we refer to Section B of the Supplementary Materials. We further remind that we follow common practice in machine learning and evaluate the performance on out-of-sample observations, that is, startups that have not been part of the training set and are thus unseen to the machine learning classifiers. In addition, we repeat the random splitting of our train and test sets five times and thus report the mean and standard deviation of our evaluation metrics across the five test sets.[4]

Overall, we find that our tailored, fused large languages model is considerably more accurate than a majority vote (i.e., a model that always predicts the majority class label) and a random vote (i.e., a model that predicts class labels randomly based on the distribution of the class labels in the training data). Both approaches represent naïve baselines from machine learning, which are outperformed by a large margin. Our tailored, fused large language model using both fundamentals and textual self-descriptions yields an AUROC of 82.78 %, a balanced accuracy of 74.33 %, and a 7.23-fold ROI. Altogether, this demonstrates the efficacy of machine learning based on our fused large language model in predicting startup success from VC platforms.

We further compare our fused large language model against common baseline text representations. Specifically, we draw upon manual feature extraction from textual data (Kaiser & Kuhn, 2020), GloVe document embeddings (Pennington et al., 2014), and a bag-of-words approach (Jurafsky & Martin, 2020). The baseline text representations have a known limitation in that they struggle with capturing long-term dependencies across language, because of which semantics are ignored to a large extent. As expected, we find that, compared to our fused large language model, the baselines are inferior. For example, the best baseline in terms of AUROC (GloVe) has a 6.41-fold ROI, while our custom, fused large language model has a 7.23-fold ROI, which is a plus of 82.19 percentage points. Note that both our fused, large language model and the bag-of-words baseline have access to the same data, that is, fundamental variables and textual self-descriptions. Hence, all performance improvements must solely be attributed to the better model architecture of our fused large language model.

In addition, we compare using fundamental variables only vs. a combination of fundamental variables and the textual self-description. Here, including textual self-description using the baseline text representations increases the AUROC by 0.62 percentage points (manual feature extraction Kaiser & Kuhn, 2020), 1.29 percentage points (GloVe Pennington et al., 2014), and 0.5 percentage points (bag-of-words). Including textual self-descriptions within our fused large language model performs best and increases the AUROC by 2.18 percentage points. As such, we yield consistent evidence that demonstrates the operational value of textual self-descriptions: a significant improvement in prediction performance is achieved by including textual self-descriptions. Altogether, this highlights the importance of textual self-descriptions for successful investing decisions.

5.2. Sensitivity to final machine learning classifier

We now provide a sensitivity analysis where we vary the final machine learning classifier (i.e., elastic net, random forest, neural network) and within our fused large language model.[5] Thereby, we confirm that our choice of a neural network for the final machine learning classifier in our fused large language model is superior. The results are reported in Table 7. By varying the final machine learning classifier in our fused large language model using both fundamentals and textual self-descriptions, we yield an AUROC of 81.76% (logistic regression), 82.51% (elastic net), 81.75% (random forest), and 82.78% (neural network). We observe a similar pattern with regard to the other performance metrics. For instance, the neural network achieves a 7.23-fold ROI. Hence, the best overall AUROC is obtained by the neural network, followed by the elastic net, logistic regression, and the random forest. Altogether, this demonstrates the efficacy of our fused large language model based on a neural network in predicting startup success from VC platforms.

5.3. Sensitivity to fine-tuning our large language model

We now experiment with fine-tuning our fused large language model. Specifically, we add a classification head to the [CLS] embedding (i.e., $o_{[CLS]}$) for classifying startup success on top of BERT. We concatenate the fundamental variables to the [CLS] embedding before feeding them to the classification head. This way, both the classification head is trained and BERT is fine-tuned simultaneously based on the task of predicting startup success. Hence, the difference to no fine-tuning lies in the fact that we now allow for parameters in BERT to be fine-tuned for the task of classification.

We fine-tuned BERT using the transformers framework from Huggingface (Wolf et al., 2020). We use a training batch size of 32 and a learning rate of $4 \cdot 10^{-5}$. We freeze the first 8 layers as they capture language patterns and an understanding of text in general. We update the weights of BERT and the classification head using the AdamW optimizer (Loshchilov & Hutter, 2018). We fine-tune for a maximum number of 5 epochs. We validate the performance every 50 steps. We performed early stopping when the loss on the hold-out set does not decrease for more than 5 steps.

Table 8 reports the results. Overall, we do not observe any performance improvement when fine-tuning our fused-large language model. The performance of fine-tuning is comparable to that of our fused large language model with a neural net classifier. Specifically, fine-tuning our fused large language model yields a 0.12 percentage point decrease in accuracy, 0.1 percentage point decrease in AUROC, and 10.64 percentage point decrease in ROI, as compared to our not fine-tuned fused large language model. Hence, the pre-trained embeddings of BERT already capture textual information relevant to the task of success prediction. Our findings underline an important aspect of machine learning: Increasing the number of trainable (or fine-tunable) parameters does not necessarily guarantee performance improvements. We discuss the finding later in Section 6.

5.4. Prediction performance across business sectors

We now perform a sensitivity analysis in which we compare how the prediction performance from our fused large language model varies across business sectors (see Table 9). In general, startup activities and outcomes vary significantly across business sectors (Konon, Fritsch, & Kritikos, 2018). For example, the sector of INFORMATION TECHNOLOGY

---

[4] We also perform an out-of-time evaluation in Section A of the Supplementary Materials, where we evaluate the performance in predicting the success of startups that originate from a period outside the one used for training. Overall, the performance remains robust but, due to the task formalization, has a smaller sample size and thus tends to have a larger variance.

[5] We also tested the performance of varying the input variables (i.e., FV, TSD, FV + TSD) within our final machine learning classifiers. We report the results in Section C.

Table 6

Prediction performance of our large language model vs. the baselines.

Approach	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
Majority vote	50.00	—†	0.00	—†	50.00	0.00	—†
Random vote	50.00	36.59	36.14	36.36	50.00	23.16	403.40
FV only	72.00	56.03	79.46	65.56	80.60	70.92	670.84
	(1.33)	(3.27)	(4.45)	(0.92)	(0.44)	(0.68)	(45.05)
FV + Manual feature extraction	72.40	56.83	78.86	65.89	81.22	71.42	681.88
	(1.15)	(3.13)	(4.65)	(0.77)	(0.39)	(0.63)	(43.12)
FV + GloVe	71.87	53.86	**85.16**	65.85	81.89	72.59	640.90
	(1.93)	(3.27)	(3.72)	(1.48)	(0.59)	(0.59)	(44.93)
FV + Bag-of-words	72.38	55.71	80.95	65.98	81.10	71.74	666.38
	(0.49)	(1.23)	(1.49)	(0.38)	(0.18)	(0.63)	(16.88)
Our fused large language model (FV + TSD)	**74.33**	**59.83**	78.28	**67.77**	**82.78**	**73.70**	**723.09**
	(0.25)	(1.79)	(2.63)	(0.15)	(0.25)	(0.49)	(24.56)

†Value not defined due to division by zero (i.e., there is no successful class).

FV = fundamental variables.

Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and model is highlighted in bold.

Table 7

Out-of-sample performance of different final machine learning classifiers within our fused large language model.

Classifier	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
Logistic Regression	73.71	58.51	78.79	67.15	81.76	72.18	704.98
	(0.40)	(0.47)	(0.48)	(0.43)	(0.31)	(0.59)	(6.43)
Elastic net	74.27	58.61	**80.43**	**67.81**	82.51	73.11	706.30
	(0.21)	(0.31)	(0.22)	(0.22)	(0.25)	(0.50)	(4.30)
Random forest	73.69	58.28	79.28	67.16	81.75	72.17	701.72
	(0.64)	(0.90)	(1.68)	(0.70)	(0.63)	(0.82)	(12.37)
Our fused large language model (neural network)	**74.33**	**59.83**	78.28	67.77	**82.78**	**73.70**	**723.09**
	(0.25)	(1.79)	(2.63)	(0.15)	(0.25)	(0.49)	(24.56)

Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and model is highlighted in bold.

Table 8

Prediction performance of our large language model with and without fine-tuning.

Fine-tuning	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
No	**74.33**	**59.83**	78.28	**67.77**	**82.78**	**73.70**	**723.09**
	(0.25)	(1.79)	(2.63)	(0.15)	(0.25)	(0.49)	(24.56)
Yes	74.21	59.06	**79.77**	67.72	82.68	73.43	712.45
	(0.79)	(2.67)	(4.84)	(0.77)	(0.35)	(0.48)	(36.77)

FV = fundamental variables, TSD = textual self-descriptions (via document embeddings).

Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and model is highlighted in bold.

typically features better data coverage and a higher number of startups, potentially leading to better predictability. Motivated by these differences, we perform a sensitivity analysis to provide insights into the extent to which textual self-descriptions contribute to performance gains across sectors. Here, we focus our evaluations on the implementation based on a neural network, i.e., the best-performing classifier. We compare high-level business sectors for easier interpretability (this is different from the fine-grained but sparse industries that are reported on Crunchbase and that we use as predictors). Overall, we find that the prediction performance is fairly robust. The AUROC varies from 72.04 % (ENERGY) to 85.39 % (INDUSTRIALS). This thus confirms that our fused large language model allows for accurate predictions across all business sectors. Furthermore, including textual self-descriptions improves the prediction performance across most business sectors. The only exceptions are the four sectors with the smallest number of data points (ENERGY, MATERIALS, REAL ESTATE, and UTILITIES). For these sectors, including textual self-descriptions does not lead to a performance improvement as compared to using only the fundamental variables for prediction. Also, the standard deviation in the prediction performance

is higher across these sectors. This implies that a sufficient number of training observations is necessary to make accurate predictions from textual self-descriptions.[6]

5.5. Prediction performance across investment events

We compare the prediction performance of our fused large language model across different events that are indicative of startup success, namely initial public offering, acquisition, and external funding. For this, we evaluate our models on subsets of the out-of-sample test

---

[6] We also tested whether the lower prediction performance in these business sectors could stem from more diverse textual self-descriptions as compared to other business sectors. However, the representations of the textual self-descriptions are (a) not more/less discriminatory for successful vs. non-successful startups across business sectors, and (b) not more/less diverse across business sectors, suggesting that more diversity in self-descriptions within specific business sectors is not a factor for lower prediction performance.

Table 9

Prediction performance across business sectors.

Business sector	N	Predictors	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
ENERGY	92	FV	**65.20** (3.50)	54.74 (6.89)	**90.23** (6.55)	**67.77** (4.82)	**77.41** (2.51)	**73.75** (2.68)	653.04 (94.75)
		FV + TSD	64.38 (3.87)	**55.56** (7.49)	79.81 (6.86)	65.23 (6.03)	72.04 (3.94)	63.88 (6.16)	**664.36** (103.05)
MATERIALS	63	FV	**71.51** (2.44)	**57.23** (5.79)	**93.59** (6.45)	**70.83** (4.83)	**81.83** (3.23)	**77.03** (4.71)	**687.33** (79.72)
		FV + TSD	67.57 (4.18)	55.35 (5.94)	83.76 (7.49)	66.5 (5.54)	78.62 (5.75)	71.73 (8.79)	661.5 (81.68)
INDUSTRIALS	1412	FV	73.00 (0.92)	61.29 (2.18)	84.57 (4.0)	70.98 (0.74)	82.85 (0.53)	76.94 (1.41)	743.17 (29.95)
		FV + TSD	**76.53** (0.77)	**65.56** (1.35)	**85.27** (1.84)	**74.11** (0.83)	**85.39** (0.41)	**79.98** (0.79)	**801.92** (18.60)
CONSUMER DISCRETIONARY	1382	FV	72.19 (2.17)	54.06 (3.64)	**77.33** (4.62)	63.44 (1.55)	80.26 (1.29)	67.63 (1.66)	643.66 (50.13)
		FV + TSD	**74.32** (1.16)	**58.19** (2.32)	75.51 (5.37)	**65.57** (1.15)	**82.52** (1.23)	**71.54** (1.15)	**700.48** (31.85)
CONSUMER STAPLES	642	FV	68.47 (1.57)	56.48 (2.4)	**83.55** (4.69)	67.28 (0.85)	78.09 (1.57)	72.5 (1.93)	677.02 (33.02)
		FV + TSD	**71.53** (2.25)	**60.52** (1.95)	81.07 (4.84)	**69.25** (2.43)	**80.70** (2.30)	**75.67** (1.63)	**732.62** (26.88)
HEALTH CARE	658	FV	63.69 (2.08)	60.67 (2.93)	**92.39** (2.14)	73.18 (1.63)	77.09 (1.4)	77.47 (2.2)	734.59 (40.37)
		FV + TSD	**66.58** (1.19)	**63.47** (2.03)	88.12 (1.46)	**73.77** (1.18)	**78.36** (0.46)	**78.88** (1.71)	**773.20** (27.91)
FINANCIALS	404	FV	70.02 (2.24)	58.92 (3.1)	**84.33** (3.73)	69.25 (1.28)	80.18 (1.18)	76.38 (1.04)	710.51 (42.59)
		FV + TSD	**72.73** (1.34)	**62.88** (2.56)	81.94 (4.55)	**71.03** (0.98)	**82.84** (1.24)	**79.25** (1.68)	**765.00** (35.20)
INFORMATION TECHNOLOGY	2202	FV	70.43 (1.43)	58.07 (2.38)	**82.09** (4.08)	67.92 (0.96)	79.72 (0.9)	73.5 (1.29)	698.91 (32.81)
		FV + TSD	**73.70** (0.73)	**62.44** (1.57)	81.21 (2.56)	**70.56** (0.90)	**82.30** (0.51)	**76.64** (0.92)	**759.02** (21.64)
COMMUNICATION SERVICES	1747	FV	72.36 (0.90)	53.36 (4.16)	**77.42** (6.12)	62.88 (1.37)	79.96 (0.72)	65.84 (1.07)	634.13 (57.22)
		FV + TSD	**73.40** (1.09)	**56.25** (2.55)	74.52 (3.25)	**64.02** (0.83)	**81.46** (0.91)	**69.22** (1.13)	**673.88** (35.07)
REAL ESTATE	236	FV	71.18 (3.56)	53.46 (3.7)	**79.24** (3.45)	**63.74** (2.66)	**81.52** (2.89)	**73.48** (3.80)	635.41 (50.87)
		FV + TSD	**71.29** (2.03)	**54.70** (3.42)	75.98 (5.79)	63.41 (1.97)	80.70 (2.33)	72.11 (2.32)	**652.46** (47.02)
UTILITIES	93	FV	64.53 (5.13)	59.6 (7.43)	**95.58** (2.67)	73.12 (4.8)	**80.26** (1.93)	**79.26** (2.12)	719.91 (102.22)
		FV + TSD	**68.64** (4.76)	**63.53** (6.34)	90.71 (3.32)	**74.57** (4.67)	78.55 (4.01)	77.47 (5.68)	**773.97** (87.28)

FV = fundamental variables, TSD = textual self-descriptions.

Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and sector is highlighted in bold.

sets split by the different events. Hence, the corresponding accuracy quantifies, for example, to what extent startups are correctly classified in the subset of startups that eventually had an initial public offering. We proceed analogously for acquisition and funding events. The results are reported in Table 10. Overall, the events vary in their frequency, as only a few startups had an initial public offering or had been acquired, whereas a larger proportion received external funding.

We find that the prediction performance is generally higher for initial public offerings and funding events. Here 82.05 % of initial public offerings and 80.17 % of funding events were predicted correctly. In contrast, only 65.54 % of acquisitions were predicted correctly, implying that, for the latter, inferences are more challenging. Again,

we confirm that machine learning benefits from incorporating textual self-descriptions. In fact, using textual self-descriptions increases the rate of correct classifications for initial public offerings by 4.02 %, for acquisitions by 3.92 %, and for funding events by 2.75 %. Therefore, our findings suggest that textual self-descriptions from VC platforms are informative for predicting startup success, consistently across all success events.

5.6. Robustness checks

We perform the following additional robustness checks. We evaluate the prediction performance of our fused large language model across

Table 10
Prediction performance across different success events.

Success event	N	Predictors	Balanced accuracy
Initial public offering	17	FV	78.03 (5.33)
		FV + TSD	**82.05** (6.84)
Acquisition	190	FV	61.62 (5.42)
		FV + TSD	**65.54** (1.36)
Funding	1244	FV	77.42 (4.09)
		FV + TSD	**80.17** (2.92)
Non-successful	2584	FV	69.23 (6.27)
		FV + TSD	**70.38** (3.10)

FV = fundamental variables, TSD = textual self-descriptions.
Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). Results are based on the neural network. The best value per success event is highlighted in bold.

different company characteristics (i. e., the age of a startup) and the length of the textual self-description. We find that the inclusion of textual self-descriptions improves the prediction performance considerably, which is consistent across startup ages and across different text lengths. This contributes to the robustness of our findings.

5.6.1. Prediction performance across startup age

The prediction performance with and without textual self-descriptions grouped across startup age is reported in Table 11. For all age groups, the majority vote and random vote as naïve baselines from machine learning are outperformed by a considerable margin and thus point towards the overall large prediction performance. In addition, all performance metrics increase by a considerable margin when including textual self-descriptions. This adds further robustness to our finding that textual self-descriptions are predictive of startup success. Furthermore, the balanced accuracy is higher for older startups with and without textual self-descriptions, indicating that more established startups potentially yield more predictive information on Crunchbase.

Varying the age of startups is also important for another reason: it allows us to assess the prediction performance across different time periods. Startups with an age between 1–12 months originate from 2015, an age between 13–24 months originate from 2014, etc. This thus contributes to the robustness of our findings.

5.6.2. Prediction performance across length of textual self-description

The prediction performance with and without textual self-descriptions across different lengths of the textual self-description is reported in Table 12. For all length groups, a majority vote and a random vote are outperformed by machine learning. In addition, a clear improvement in AUROC is found for all groups when including textual self-descriptions. Overall, this adds robustness to our finding that textual self-descriptions are predictive of startup success. Furthermore, the AUROC is higher for startups with longer textual self-descriptions. Similarly, both metrics increase for the baseline without textual self-descriptions. Still, the length of the textual self-description appears to play a minor role in the prediction performance.

5.7. Post-hoc explainability of our machine learning approach

The previous analyses demonstrate the performance improvement of including textual self-descriptions for the task of startup success prediction. Now, we analyze the contributions of each variable (i. e., fundamentals and textual self-descriptions) for predicting startup success. To this end, we aim to understand how our fused large language model uses the variables to arrive at predictions. We use the SHAP value method (Lundberg & Lee, 2017). Intuitively, the SHAP value method treats the prediction of a model as a cooperative game, i.e., the prediction (i.e., the payoff) must be allocated fairly among the feature values (i.e., the individual players) based on their contribution. Hence, the SHAP value method enables a nuanced understanding of how each feature contributes to the prediction of the model and is frequently used for understanding machine learning in management applications (e.g., Senoner, Netland, & Feuerriegel, 2022).

SHAP values are computed for each observation separately, i.e., every feature within the vector of each observation is assigned a SHAP value. SHAP values can also be interpreted at the model level. Therefore, we quantify both feature attribution and feature importance based on the SHAP values. Feature attribution is directly determined by the SHAP values and feature importance is computed by averaging the absolute SHAP values across observations. We follow previous research (Senoner et al., 2022) and aggregate the SHAP values (sum) across the document embedding of the TSD to one feature representation.

Fig. 3 shows the summary plot of SHAP values computed for the predictions of our fused large language model. In the left plot, the dots across each feature represent the feature attribution for each prediction of a specific observation. The right plot shows the mean of the absolute SHAP values across all samples. Both plots show the 20 features with the highest computed feature importance, ranking them from highest (top) to lowest (bottom) importance. Notably, the aggregated representation of the textual self-description is the most important feature, indicating that it contributes, on average, the most to the prediction of our fused large language model. Here feature attributions range from −0.53 to 0.76 with a mean absolute value of 0.29. Thus, out of all features, the textual self-description adds the most to the predictions of our fused large language model.

Variables characterizing the momentum of a startup also make important contributions to the predictions of our fused large language model. Overall, the age of the startup is the second most important feature. Feature attributions range from −0.28 for a startup age of 35.59 months to 0.20 for a startup age of 4 months. On average, a higher age of startups is estimated to have a negative feature attribution to success prediction. This may indicate that startups with a longer market presence face reduced probabilities of success due to, for example, lower perceived growth potential and questionable viability of their business model if they have not yet achieved success. In addition, recent funding activities (last round time lag) are estimated as positive contributions to predictions of success, with feature attributions ranging from −0.37 to 0.05. A plausible explanation might be that recent funding signals reduced risk, as other investors have recently found the startup promising enough to invest in (i. e., a form of validation). Similarly, the number of investors in the last investment round contributes positively to the success predictions, reinforcing the idea that previous funding may serve as a form of validation that predicts success also in the future.

Founder characteristics also play an important role in the predictions made by our fused large language model. Among these, the number of founders and their educational backgrounds are highly influential. Here, the total number of founders positively contributes to success predictions, which suggests that startups with a higher number of founders may benefit from diverse skill sets and shared responsibilities. The total number of degrees among the founders also shows a positive contribution, with attributions ranging from −0.09

Table 11
Prediction performance across startup age.

Startup age	N	Predictors	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
1–12 months	1174	FV	66.99 (1.56)	58.82 (2.44)	**88.09** (2.86)	70.47 (1.18)	78.21 (1.25)	75.34 (1.36)	709.14 (33.51)
		FV + TSD	**71.66** (0.73)	**64.55** (2.44)	83.61 (4.01)	**72.76** (0.93)	**81.20** (1.06)	**78.63** (0.61)	**788.05** (33.53)
13–24 months	1307	FV	70.44 (1.93)	54.77 (3.25)	**79.33** (4.78)	64.68 (2.01)	79.64 (0.75)	70.76 (1.28)	653.53 (44.75)
		FV + TSD	**73.81** (0.85)	**59.62** (2.29)	78.71 (3.2)	**67.78** (1.25)	**82.54** (1.09)	**74.40** (1.38)	**720.26** (31.50)
25–36 months	1460	FV	72.68 (1.53)	53.9 (5.23)	67.43 (8.08)	59.41 (2.16)	80.52 (0.82)	63.87 (0.59)	641.56 (71.9)
		FV + TSD	**74.05** (1.25)	**54.04** (1.99)	**70.24** (1.88)	**61.07** (1.67)	**82.15** (1.08)	**65.13** (1.89)	**643.50** (27.44)

FV = fundamental variables, TSD = textual self-descriptions.
Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and age group is highlighted in bold.

Table 12
Prediction performance for different lengths of the textual self-descriptions.

Text length	N	Predictors	Balanced accuracy	Precision	Recall	F_1-score	AUROC	AUCPR	ROI
50–100 words	2306	FV	71.05 (1.64)	58.11 (3.79)	79.47 (4.43)	66.94 (1.11)	79.82 (0.7)	73.0 (0.72)	699.47 (52.15)
		FV + TSD	**73.55** (0.54)	**60.85** (1.55)	**80.59** (2.21)	**69.31** (0.63)	**82.29** (0.71)	**75.86** (1.25)	**737.08** (21.28)
101–200 words	1103	FV	73.56 (1.36)	54.42 (2.63)	**81.26** (4.10)	65.06 (0.91)	82.15 (0.69)	70.65 (1.6)	648.61 (36.25)
		FV + TSD	**75.23** (0.81)	**59.49** (2.85)	76.32 (3.12)	**66.78** (1.46)	**83.36** (0.85)	**71.90** (1.55)	**718.42** (39.23)
≥201 words	279	FV	**75.93** (2.65)	38.74 (4.54)	**76.80** (7.95)	51.2 (3.88)	83.36 (2.08)	**55.01** (9.07)	432.95 (62.45)
		FV + TSD	73.46 (3.18)	**49.67** (7.84)	59.47 (7.73)	**53.52** (4.87)	**84.04** (2.67)	53.51 (5.41)	**583.33** (107.92)

FV = fundamental variables, TSD = textual self-descriptions.
Note: Reported is the mean (and standard deviation) out-of-sample prediction performance across 5 random splits (in %). The best value per metric and length group is highlighted in bold.

to 0.09. This suggests that a higher number of educational degrees within the founding team may predict success, possibly reflecting the founding team's capability to tackle complex challenges and innovate. In addition, the presence of a LinkedIn profile (and an email) for founders also stands out as an important and positive contributor to the predictions of our model. This indicates that visible professional networking and the credibility it brings might be a strong predictor of later startup success.

The contributions of the sectors in which a startup operates are also reflected in the SHAP values. For example, as seen by the SHAP values, SOFTWARE is the most influential sector feature. Startups in the SOFTWARE sector typically exhibit high growth potential, so that the information of whether a startup operates in this sector helps to predict later success. Similarly, sectors such as HEALTHCARE and ARTIFICIAL INTELLIGENCE also show positive contributions, with HEALTHCARE ranging from −0.04 to 0.13 and ARTIFICIAL INTELLIGENCE from −0.04 to 0.22. These sectors are typically characterized by innovative solutions of high impact. In contrast, sectors such as FOOD AND BEVERAGE and HARDWARE show smaller but still positive contributions, which could be attributed to higher capital requirements and longer time to market.

6. Discussion

6.1. Managerial implications

Our work demonstrates that VC platforms can be used to predict startup success and thus support investing decisions. We find that

predictions from our fused large language model achieve an AUROC of up to 82.78 %, a balanced accuracy of up to 74.33 %, and a 7.33-fold ROI. Thereby, baselines without machine learning (e. g., a majority vote) are outperformed by a considerable margin. Prior literature has already shown that various fundamental variables are predictors of startup success, whereas we show that additional predictive power is offered by textual self-descriptions. Here, we find that incorporating textual self-descriptions through our fused large language model increases the AUROC by 2.18 percentage points, the balanced accuracy by 2.33 percentage points, and the ROI by 52.25 percentage points. The increase in prediction performance is statistically significant. As such, our work is of direct managerial relevance as it provides computerized decision support for venture capitalists with the prospect of making financially rewarding investments.

We also show that traditional machine learning methods for making predictions from text (e.g., bag-of-words with manual feature extraction (Kaiser & Kuhn, 2020)) are inferior to state-of-the-art methods based on large language models. Traditional methods (e.g., Kaiser & Kuhn, 2020) rely on manually crafting features from text that might not capture the entire latent textual information. In contrast, our fused large language model utilizes so-called neural representation learning, capturing latent information in texts through an automated, data-driven procedure that learns from data. Notably, we observe that fine-tuning the language model does not increase the performance. The complexity behind the alignment of pre-trained knowledge and target domain characteristics has been discussed in recent NLP literature (Bertsch
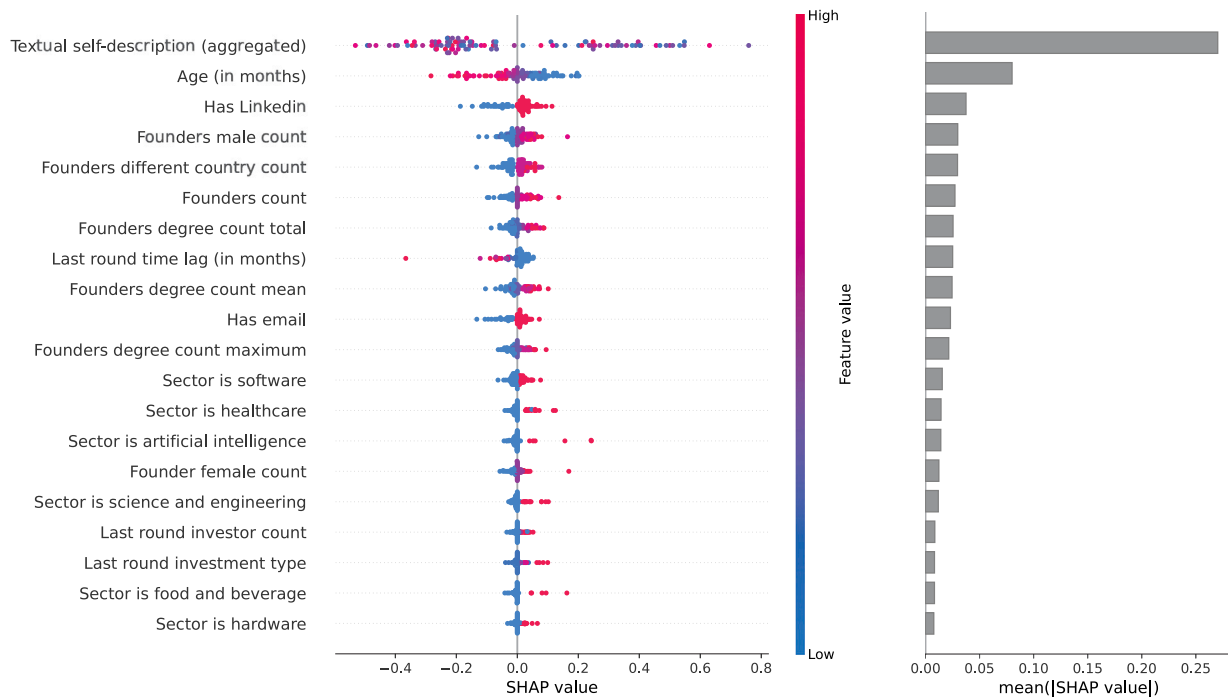
Fig. 3. Left: Summary plot of the SHAP values for our fused large language model predictions. Each dot represents a SHAP value for a feature across different samples, where color indicates a high (red) to low (blue) feature value. Right: Bar plot of the mean (absolute) of SHAP values across all samples. The features are sorted by their feature importance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

et al., 2024; Zhang, Liu, Cherry, & Firat, 2024), where evidence is provided that fine-tuning does not always help performance due to it being highly task- and data-specific. The fact that fine-tuning shows similar performance as no fine-tuning underlines an important aspect of machine learning: increasing the number of trainable or fine-tunable parameters does not necessarily guarantee performance improvements. Thus managers should carefully consider the use of large language models when dealing with decision problems that involve text data.

The improvements in prediction performance when incorporating textual self-descriptions are robust across all business sectors and economically significant. To assess the practical implications, we translate the prediction performance into investment portfolio performance (ROI). Our results show significantly increased ROI when incorporating textual self-descriptions through our fused large language model: The best-performing baseline without textual self-descriptions amounts to a 6.71-fold ROI, while our fused large language model achieves a 7.23-fold ROI. The financial gains from our fused large language model can be further explained by the substantial costs of false positives in the context of startup investment decisions. False positive classifications for investment decisions lead to investing in startups predicted to succeed but ultimately failing. Hence, investments in startups that eventually fail lead to a potential loss of the entire investment amount. Our model significantly reduces the probability of false positives compared to the baselines, thereby increasing the overall returns from our machine learning approach for making investment decisions.

The Crunchbase database is widely used for academic research, which in turn yields practical implications. Crunchbase offers an online platform with comprehensive data on startups including fundamental variables (e.g., the age of the startup) and textual self-descriptions. Such data has key differences from the data traditionally collected by VC investors for decision-making (Kaplan & Lerner, 2016; Retterath & Braun, 2020). Here, two reasons stand out why investors traditionally have only little data about startup trajectories. On the one hand, investors typically collect only a few variables about startups (e.g., via scorecards) (Böhm et al., 2017; Yankov et al., 2014) and

often not in a structured format (C.F.A. Institute, 2015). On the other hand, and more importantly, investors typically screen only a few dozen startups and thus only have access to startup data for a very small sample size (Retterath & Kavadias, 2020), which precludes data-driven inferences. In sum, both of the aforementioned reasons are salient hurdles for training and deploying machine learning tools. As a remedy, prior literature evaluated the predictive ability of fundamental variables on Crunchbase (e.g., Arroyo et al., 2019; Dellermann et al., 2017; Sharchilev et al., 2018). We add to prior literature by using large language models to incorporate the additional predictive ability of textual self-descriptions on Crunchbase. Hence, Crunchbase offers valuable data for VC investors and other practitioners regarding the evaluation of startups and the enhancement of decision-making tools.

6.2. Methodological implications

We contribute to business analytics research by demonstrating the operational value of large language models in the context of more effective investment decisions. Thereby, we connect to a growing stream of machine learning in business analytics (e.g., Bastani et al., 2022; Choi et al., 2018; Cohen, 2018; Mišić & Perakis, 2020). Different from explanatory analysis (i.e., regression analysis) that merely estimates associations in an in-sample setting, machine learning is concerned with how well inferences can be made in an out-of-sample setting. Here, we demonstrate an impactful application of machine learning in VC decision-making.

Large language models have several favorable advantages over traditional methods for natural language processing. On the one hand, large language models provide a flexible way to capture semantics and structure in textual materials, thereby bolstering the prediction performance over alternative machine learning approaches (e.g., bag-of-words). On the other hand, large language models can learn from vast amounts of unlabeled texts through pre-training. As such, large language models can often be applied out-of-the-box with little need for fine-tuning. This is beneficial as it greatly reduces the manual effort and

the cost for data annotation. However, applications of large language models in business analytics are still rare, while we develop a tailored, *fused* architecture for our decision-making problem. As shown above, large language models may need custom tailoring. In our case, we build a *fused* large language model that can leverage running text but where the final prediction layer can also process structured data. As such, we expect that our *fused* large language model is of direct relevance for many business analytics settings where the goal is to expand traditional operational information in structured form with additional text data.

Our study offers implications for the use of large language models in business analytics. We based our predictions on a tailored large language model, a recent innovation from machine learning research. We expect that large language models are beneficial for a wide array of managerial decision-making tasks. This opens new opportunities for research by adapting large language models to, for instance, sales and demand forecasting from social media data, credit scoring, and business failure prediction.

6.3. Limitations and future research

As with other works, ours is not free of limitation, which offers promising directions for future research. First, large language models such as BERT may embed biases that are populated in downstream tasks. Large language models are trained on vast corpora of text data, which inevitably contain societal biases (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016; Caliskan, Bryson, & Narayanan, 2017; Garg, Schiebinger, Jurafsky, & Zou, 2018). Consequently, there is a risk that these embedded biases could influence predictions (De-Arteaga, Feuerriegel, & Saar-Tsechansky, 2022), potentially disadvantaging certain startups. Addressing this challenge requires ongoing efforts to mitigate biases within large language models. Future research could focus on refining these models to ensure equitable decision-making processes. For now, we call for careful use when deploying our model in practice. Second, our work is centered on data from the VC platform Crunchbase. While this choice is informed by prior research (Arroyo et al., 2019; Dellermann et al., 2017; Sharchilev et al., 2018), it does introduce a limitation to our work. Crunchbase is a leading online VC platform that collects rich startup and investor data; however, it may not capture the full set of startups and investors globally. Future work might expand the data sources to include a broader spectrum of startups, enhancing the relevance and robustness across different sectors and regions. Third, the economic landscape of startups is dynamically evolving. To ensure ongoing predictive performance, continuous data collection and model retraining is needed. Lastly, the dynamic nature of the economic landscape might lead to startups adapting their textual self-descriptions in response to model predictions. This suggests an area for future research on the equilibrium implications of textual self-descriptions and model predictions. Specifically, analyzing equilibria could unveil the response of startups to prediction models in designing self-descriptions. Such analysis would require a different form of analysis using equilibria but not machine learning as in our paper.

7. Conclusion

The majority of startups fail. Owing to this, the decision-making of investors is confronted with considerable challenges in identifying which startups will turn out to be successful. To support investors in this task, we developed a tailored, *fused* large language model that incorporates the textual self-description of startups alongside other fundamental variables to predict startup success. Here, we show that additional predictive power is offered by the textual self-descriptions. Our model helps investors identify investment targets that promise financial returns. For this, our work provides computerized decision support that allows investors to automate their screening process with data-driven technologies. Furthermore, our study highlights the potential of applying large language models in domains where relevant text

data is available but has not traditionally been used for predicting outcomes needed for decision-making. For example, similar to ours, future work could attempt using textual self-descriptions of venture capitalists to predict the performance of their investments. In such scenarios, the findings from our study suggest that combining textual information with conventional data sources may have the potential to significantly enhance predictive accuracy and decision-making processes.

CRediT authorship contribution statement

Acknowledgments

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2024.09.011.

References

Aggarwal, R., & Singh, H. (2013). Differential influence of blogs across different stages of decision making: The case of venture capitalists. *MIS Quarterly*, *37*(4), 1093–1112.

Alamsyah, A., & Nugroho, T. B. A. (2018). Predictive modelling for startup and investor relationship based on crowdfunding platform data. *Journal of Physics: Conference Series*, *971*(Article 012002).

Arroyo, J., Corea, F., Jimenez-Diaz, G., & Recio-Garcia, J. A. (2019). Assessment of machine learning performance for decision support in venture capital investments. *IEEE Access*, *7*, 124233–124243.

Bastani, H., Zhang, D., & Zhang, H. (2022). Applied machine learning in operations management. In V. Babich, J. Birge, & G. Hilary (Eds.), *Innovative Technology at the Interface of Finance and Operations* (pp. 189–222). Cham, Switzerland: Springer.

Baum, J. A. C., & Silverman, B. S. (2004). Picking winners or building them? Alliance, intellectual, and human capital as selection criteria in venture financing and performance of biotechnology startups. *Journal of Business Venturing*, *19*(3), 411–436.

Bertsch, A., Ivgi, M., Alon, U., Berant, J., Gormley, M. R., & Neubig, G. (2024). In-context learning with long-context models: An in-depth exploration. arXiv preprint arXiv:2405.00200.

Böhm, M., Weking, J., Fortunat, F., Müller, S., Welpe, I., & Krcmar, H. (2017). The business model DNA: Towards an approach for predicting business model success. In *Internationale Tagung Wirtschaftsinformatik*.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Borchert, P., Coussement, K., De Caigny, A., & De Weerdt, J. (2023). Extending business failure prediction models with textual website content using deep learning. *European Journal of Operational Research*, *306*(1), 348–357.

Butler, J. S., Garg, R., & Stephens, B. (2020). Social networks, funding, and regional advantages in technology entrepreneurship: An empirical analysis. *Information Systems Research*, *31*(1), 198–216.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186.

C. F. A. Institute (2015). The herding mentality: Behavioral finance and investor biases. URL https://blogs.cfainstitute.org/investor/2015/08/06/the-herding-mentality-behavioral-finance-and-investor-biases/.

Chang, S. J. (2004). Venture capital financing, strategic alliances, and the initial public offerings of internet startups. *Journal of Business Venturing*, *19*(5), 721–741.

Choi, T.-M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management*, *27*(10), 1868–1883.

CNN (2020). *Facebook's first big investor, Peter Thiel, cashes out*. Cable News Network, URL https://money.cnn.com/2012/08/20/technology/facebook-peter-thiel/index.html.

Cohen, M. C. (2018). Big data and service operations. *Production and Operations Management*, *27*(9), 1709–1723.

Connelly, B. L., Certo, S. T., Ireland, R. D., & Reutzel, C. R. (2011). Signaling theory: A review and assessment. *Journal of Management*, *37*(1), 39–67.

Conti, A., & Graham, S. J. H. (2020). Valuable choices: Prominent venture capitalists' influence on startup CEO replacements. *Management Science*, *66*(3), 1325–1350.

Cui, R., Gallino, S., Moreno, A., & Zhang, D. J. (2018). The operational value of social media information. *Production and Operations Management*, *27*(10), 1749–1769.

De-Arteaga, M., Feuerriegel, S., & Saar-Tschansky, M. (2022). Algorithmic fairness in business analytics: Directions for research and practice. *Production and Operations Management*, *31*(10), 3749–3770.

De Bock, K. W., Coussement, K., & Lessmann, S. (2020). Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach. *European Journal of Operational Research*, *285*(2), 612–630.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, *269*(2), 760–772.

Dellermann, D., Lipusch, N., Ebel, P., Popp, K. M., & Leimeister, J. M. (2017). Finding the unicorn: Predicting early stage startup success through a hybrid intelligence method. In *Internation Conference on Information Systems*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint, arXiv: 1810.04805.

Feuerriegel, S., & Gordon, J. (2019). News-based forecasts of macroeconomic indicators: A semantic path model for interpretable predictions. *European Journal of Operational Research*, *272*(1), 162–175.

Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, *66*(1), 111–126.

Feuerriegel, S., Shrestha, Y. R., von Krogh, G., & Zhang, C. (2022). Bringing artificial intelligence to business management. *Nature Machine Intelligence*, *4*(7), 611–613.

Fleximize (2023). The speed of a unicorn. URL https://fleximize.com/unicorns/.

Forbes (2019). 10 Tips for successfully scaling your service-based business. URL https://www.forbes.com/sites/theyec/2019/10/29/10-tips-for-successfully-scaling-your-service-based-business/.

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(16), E3635–E3644.

van Gelderen, M., Thurik, R., & Bosma, N. (2005). Success and risk factors in the pre-startup phase. *Small Business Economics*, *24*(4), 365–380.

Geuens, S., Coussement, K., & De Bock, K. W. (2018). A framework for configuring collaborative filtering-based recommendations derived from purchase data. *European Journal of Operational Research*, *265*(1), 208–218.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2). Springer.

Haupt, J., Bender, B., Fabian, B., & Lessmann, S. (2018). Robust identification of email tracking: A machine learning approach. *European Journal of Operational Research*, *271*(1), 341–356.

Hegde, D., & Tumlinson, J. (2014). Does social proximity enhance business partnerships? Theory and evidence from ethnicity's role in U.S. venture capital. *Management Science*, *60*(9), 2355–2380.

Ho, T. K. (1995). Random decision forests. In *International Conference on Document Analysis and Recognition*.

Holmes, P., Hunt, A., & Stone, I. (2010). An analysis of new firm survival using a hazard function. *Applied Economics*, *42*(2), 185–195.

Huang, L., & Pearce, J. L. (2015). Managing the unknowable: The effectiveness of early-stage investor gut feel in entrepreneurial investment decisions. *Administrative Science Quarterly*, *60*(4), 634–670.

Jakubik, J., & Feuerriegel, S. (2022). Data-driven allocation of development aid toward sustainable development goals: Evidence from HIV/AIDS. *Production and Operations Management*, *31*(6), 2739–2756.

Jin, F., Wu, A., & Hitt, L. (2017). Social is the new financial: How startup social media activity influences funding outcomes. In *Academy of Management Proceedings*.

Jurafsky, D., & Martin, J. H. (2020). Speech and language processing, (3rd edition). Stanford.

Kadar, C., Maculan, R., & Feuerriegel, S. (2019). Public decision support for low population density areas: An imbalance-aware hyper-ensemble for spatio-temporal crime prediction. *Decision Support Systems*, *119*, 107–117.

Kaiser, U., & Kuhn, J. M. (2020). The value of publicly available, textual and non-textual information for startup performance prediction. *Journal of Business Venturing Insights*, *14*, Article e00179.

Kaplan, S. N., & Lerner, J. (2016). Venture capital data: Opportunities and challenges. *Measuring entrepreneurial businesses: Current knowledge and challenges*, 413–431.

Katila, R., Chen, E. L., & Piezunka, H. (2012). All the right moves: How entrepreneurial firms compete effectively. *Strategic Entrepreneurship Journal*, *6*(2), 116–132.

Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.-C., & Johnson, J. E. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research*, *283*(1), 217–234.

Konon, A., Fritsch, M., & Kritikos, A. S. (2018). Business cycles and start-ups across industries: An empirical analysis of german regions. *Journal of Business Venturing*, *33*(6), 742–761.

Kozodoi, N., Jacob, J., & Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, *297*(3), 1083–1094.

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, *104*, 38–48.

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, *281*(3), 628–641.

Kriebel, J., & Stitz, L. (2022). Credit default prediction from user-generated text in peer-to-peer lending using deep learning. *European Journal of Operational Research*, *302*(1), 309–323.

Lau, R. Y. K., Zhang, W., & Xu, W. (2018). Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production and Operations Management*, *27*(10), 1775–1794.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136.

Littunen, H., & Niittykangas, H. (2010). The rapid growth of young firms during various stages of entrepreneurship. *Journal of Small Business and Enterprise Development*, *17*(1), 317–350.

Loshchilov, I., & Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*.

Lussier, R. N. (1995). A nonfinancial business success versus failure prediction model. *Journal of Small Business Management*, *33*(1), 8–20.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research*, *261*(2), 656–665.

Mann, R. J., & Sager, T. W. (2007). Patents, venture capital, and software start-ups. *Research Policy*, *36*(2), 193–208.

McKenzie, D., & Sansone, D. (2017). Man vs. machine in predicting successful entrepreneurs: Evidence from a business plan competition in Nigeria. *World of Banking*, *8271*, 5.

Mišić, V. V., & Perakis, G. (2020). Data analytics in operations management: A review. *Manufacturing & Service Operations Management*, *22*(1), 158–169.

Nahata, R. (2008). Venture capital reputation and investment performance. *Journal of Financial Economics*, *90*(2), 127–151.

Nanda, R., & Rhodes-Kropf, M. (2013). Investment cycles and startup innovation. *Journal of Financial Economics*, *110*(2), 403–418.

Naumzik, C., Feuerriegel, S., & Weinmann, M. (2022). I will survive: Predicting business failures from customer ratings. *Marketing Science*, *41*(1), 188–207.

Ozyurt, Y., Hatt, T., Zhang, C., & Feuerriegel, S. (2022). A deep Markov model for clickstream analytics in online shopping. In *Proceedings of the ACM Web Conference*.

Parhankangas, A., & Ehrlich, M. (2014). How entrepreneurs seduce business angels: An impression management approach. *Journal of Business Venturing*, *29*(4), 543–564.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ragothaman, S., Naik, B., & Ramakrishnan, K. (2003). Predicting corporate acquisitions: An application of uncertain reasoning using rule induction. *Information Systems Frontiers*, *5*(4), 401–412.

Retterath, A., & Braun, R. (2020). Benchmarking venture capital databases. Available at SSRN 3706108.

Retterath, A., & Kavadias, S. (2020). How to hit home runs: Portfolio strategies and returns in formal and informal venture capital. Available at SSRN 3527412.

Scott, E. L., Shu, P., & Lubynsky, R. M. (2020). Entrepreneurial uncertainty and expert evaluation: An empirical analysis. *Management Science*, *66*(3), 1278–1299.

Senoner, J., Kratzwald, B., Kuzmanovic, M., Netland, T. r. H., & Feuerriegel, S. (2023). Addressing distributional shifts in operations management: The case of order fulfillment in customized production. *Production and Operations Management*, *32*(10), 3022–3042.

Senoner, J., Netland, T. r., & Feuerriegel, S. (2022). Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, *68*(8), 5704–5723.

Sharchilev, B., Roizner, M., Rumyantsev, A., Ozornin, D., Serdyukov, P., & de Rijke, M. (2018). Web-based startup success prediction. In *International Conference on Information and Knowledge Management*.

Stevenson, M., Mues, C., & Bravo, C. (2021). The value of text for small business default prediction: A deep learning approach. *European Journal of Operational Research*, *295*(2), 758–771.

Täuscher, K., & Kietzmann, J. (2017). Learning from failures in the sharing economy. *Science Technology*, *67*(9), 2047–2059.

Toetzke, M., Banholzer, N., & Feuerriegel, S. (2022). Monitoring global development aid with machine learning. *Nature Sustainability*, *5*(6), 533–541.

U. S. Bureau of Labor Statistics (2016). Business employment dynamics: Entrepreneurship and the U.S. economy. URL https://www.bls.gov/bdm/entrepreneurship/entrepreneurship.htm.

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, *238*(2), 505–513.

Wei, C.-P., Jiang, Y.-S., & Yang, C.-S. (2008). Patent analysis for supporting merger and acquisition (M&A) prediction: A data mining approach. In *Workshop on E-Business*.

Weibl, J., & Hess, T. (2019). Finding the next unicorn: When big data meets venture capital. In *Hawaii International Conference on System Sciences*.

Weking, J., Böttcher, T. P., Hermes, S., & Hein, A. (2019). Does business model matter for startup success? A quantitative analysis. In *European Conference on Information Systems*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2020). Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yankov, B., Ruskov, P., & Haralampiev, K. (2014). Models and tools for technology start-up companies success analysis. *Economic Alternatives*, *3*, 15–24.

Zhang, B., Liu, Z., Cherry, C., & Firat, O. (2024). When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *International Conference on Learning Representations (ICLR)*.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *67*(2), 301–320.