Exploiting Expert Knowledge for Assigning Firms to Industries: A Novel Deep Learning Method

Xiaohang Zhao[1], Xiao Fang[2,*], Jing He[2], Lihua Huang[3]

[1] School of Information Management & Engineering,
Shanghai University of Finance and Economics, Shanghai, China

[2] Lerner College of Business and Economics, University of Delaware, Newark, DE, USA

[3] School of Management, Fudan University, Shanghai, China

∗ Corresponding Author: Xiao Fang, xfang@udel.edu

Abstract: Industry assignment, which assigns firms to industries according to a predefined Industry Classification System (ICS), is fundamental to a large number of critical business practices, ranging from operations and strategic decision making by firms to economic analyses by government agencies. Three types of expert knowledge are essential to effective industry assignment: definition-based knowledge (i.e., expert definitions of each industry), structure-based knowledge (i.e., structural relationships among industries as specified in an ICS), and assignment-based knowledge (i.e., prior firm-industry assignments performed by domain experts). Existing industry assignment methods utilize only assignment-based knowledge to learn a model that classifies unassigned firms to industries, and overlook definition-based and structure-based knowledge. Moreover, these methods only consider which industry a firm has been assigned to, but ignore the time-specificity of assignment-based knowledge, i.e., when the assignment occurs. To address the limitations of existing methods, we propose a novel deep learning-based method that not only seamlessly integrates the three types of knowledge for industry assignment but also takes the time-specificity of assignment-based knowledge into account. Methodologically, our method features two innovations: dynamic industry representation and hierarchical assignment. The former represents an industry as a sequence of time-specific vectors by integrating the three types of knowledge through our proposed temporal and spatial aggregation mechanisms. The latter takes industry and firm representations as inputs, computes the probability of assigning a firm to different industries, and assigns the firm to the industry with the highest probability. We conduct extensive evaluations with two widely used ICSs and demonstrate the superiority of our method over prevalent existing methods.

Keywords: Financial technology (Fintech), industry assignment, deep learning, industry classification system (ICS), hierarchical classification, label embedding

1. Introduction

Fostered by the rapid advancement of Information Technology (IT), Financial Technology (Fintech) has attracted increasing research attention from the business field in general and the Information Systems (IS) field in particular (Hendershott et al. 2017, Goldstein et al. 2019). Generally speaking, Fintech refers to the development of IT-based solutions to solve important financial problems with the goal of making financial services and business practices more efficient and effective (Hendershott et al. 2017, Goldstein et al. 2019). Over the years, IS researchers have tackled important financial problems ranging from discovering firms' financial risks to measuring firms' dyadic business proximity, using advanced IT solutions, especially solutions based on artificial intelligence and machine learning (Bao and Datta 2014, Shi et al. 2016). One important financial problem is to assign firms to industries according to a predefined Industry Classification System (ICS), namely the industry assignment problem (Wood et al. 2017). According to Hoberg and Phillips (2016), ICSs that define industry boundaries and industry competitiveness are central to business and economics research. There are two streams of research dedicated to the design of ICSs (Wood et al. 2017). One stream develops new ways of grouping firms and designs novel structures of ICSs that are different from widely used existing ones. The other stream, including our study, solves the industry assignment problem for existing ICSs.

An ICS is a taxonomy of business activities that aims to group firms operating similar lines of business into the same categories (Phillips and Ormsby 2016). The industry categories of an ICS are typically organized hierarchically as a tree, each node of which corresponds to an industry and is labeled by a numeric code as well as a title of the industry. The granularity of the classification increases from the top level of the tree to the bottom level. One prominent ICS is the North American Industry Classification System (NAICS), which was developed in collaboration by the Canadian, Mexican, and U.S. governments, and is widely used by government agencies, business practitioners, and academic researchers (Bhojraj et al. 2003, Phillips and Ormsby 2016). Figure 1 provides an excerpt from the NAICS hierarchy that highlights the "Direct Life Insurance Carriers" industry, which is coded as 524113.[1] As shown, the NAICS is comprised of five levels of industries. A root node is placed at level zero to denote the entire ICS. From the first level to the fifth level, the industry codes have two, three, four, five, and six digits, respectively. Given a target ICS, a focal level of interest, and the firms to be assigned, the objective of the industry assignment problem is to assign each firm to a focal-level industry of the target ICS that best covers the firm's business activities. As an example, when the target ICS is the NAICS (Figure 1) and the focal level is two,

[1]The tree is based on the NAICS taxonomy as revised in 2012. See https://www.census.gov/naics/?58967?yearbck=2012 for the complete hierarchy.

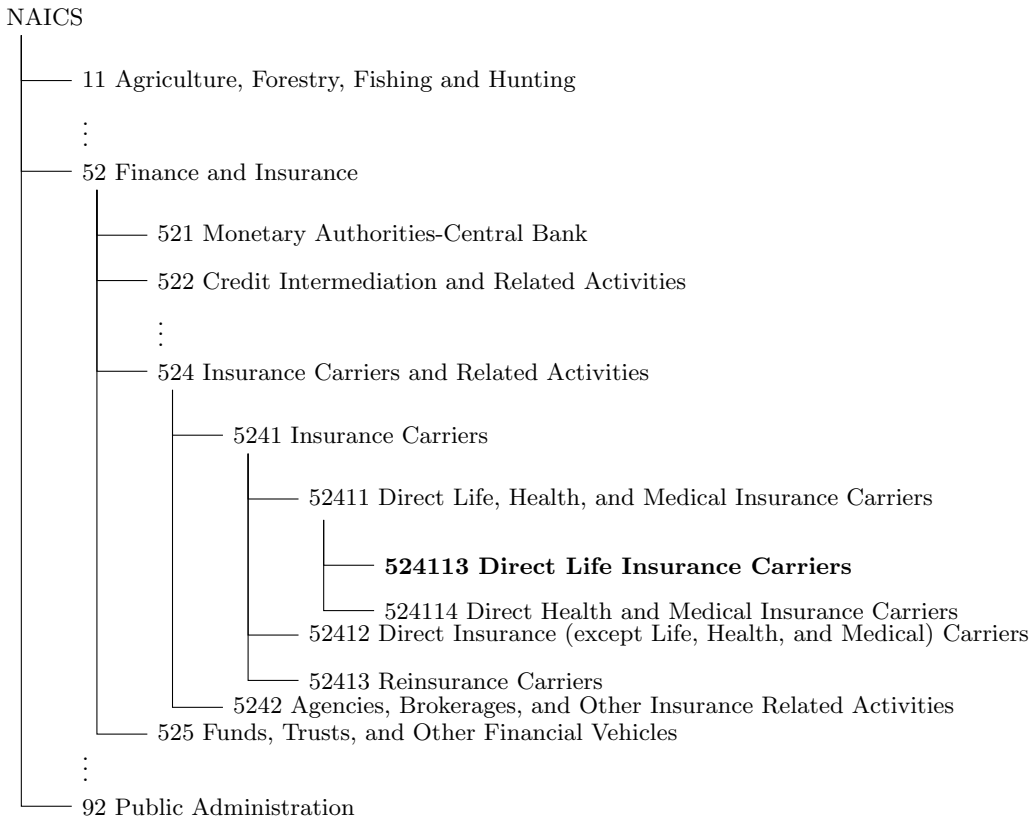

Figure 1 A Partial View of the NAICS

each firm needs to be assigned to one of the 99 NAICS second-level industries (e.g., 521 [Monetary Authorities-Central Bank]).

Industry assignment is necessary for an expanding array of important business practices. For firms, knowing the industry to which a firm belongs is important for its daily operations and strategic decisions. For example, a firm needs to know its industry code to determine whether it is qualified for certain tax deduction programs or eligible for bidding on government contracts that are only offered to specific industries.[2] A firm also frequently needs to identify its industry peers, i.e., those firms that are assigned to the same industry as itself. These peers serve as references for the firm when making a series of strategic decisions, such as shaping corporate policies (Cao et al. 2019) and determining executive compensation (Bizjak et al. 2011). For investors, industry assignment information is often used for relative evaluation (e.g., Goodman and Peavy 1983), where the stock value of a firm is evaluated within the context of its industry peers. For government agencies, industry assignment information has been used for collecting and analyzing economic data. As an example, the U.S. Bureau of Labor Statistics reports employment and output data for firms grouped

[2]Please visit https://www.census.gov/naics/ and then click on the "FAQs" tab.

by their two-digit NAICS codes,[3] which shed light on how technology development impacts labor productivity by industry.[4] Industry assignment also plays an important role in business research. Researchers often employ industry assignment information to identify economically comparable firms from the same industry as a control group, restrict samples to specific industries, develop industry dummies to control for fixed effects, or detect industry effects (Kahle and Walkling 1996, McGahan and Porter 1997, Cavaglia et al. 2000, Weiner 2005). Moreover, it is common to utilize industry assignment information for several of the purposes listed above in a single study, e.g., Ahern and Harford (2014) and Bonaime et al. (2018).[5]

Given the important role of industry assignment in business practices and research, substantial efforts have been devoted to it. Traditionally, industry assignment was partially a manual procedure conducted by human experts from authorities such as the U.S. Census Bureau, which is costly and time-consuming (Kearney and Kornbau 2005, Gweon et al. 2017). Consequently, attempts have been made to develop methods for automatic industry assignment. Such methods typically learn a classification model from prior assignment cases performed by human experts and apply the learned model to automatically classify unassigned firms (Kearney and Kornbau 2005, Rodrigues et al. 2012, Wood et al. 2017). For example, Kearney and Kornbau (2005) classify firms into NAICS industries with keyword matching techniques, while Wood et al. (2017) build a deep learning-based classification model that takes texts describing a firm as input and predicts the firm's NAICS industry.

Three types of expert knowledge are essential for effective industry assignment: definition-based, structure-based, and assignment-based knowledge. These types of knowledge are created by domain experts, who design an ICS or assign firms to industries according to an ICS, and they inform automatic industry assignment in different respects. Definition-based knowledge for an industry is a paragraph describing the business activities covered by the industry. As an example, the textual definition for the NAICS industry "Direct Life Insurance Carriers" (coded as 524113 in Figure 1), is

> *This U.S. industry comprises establishments primarily engaged in initially underwriting (i.e., assuming the risk and assigning premiums) annuities and life insurance policies, disability income insurance policies, and accidental death and dismemberment insurance policies.* [6]

[3] https://www.bls.gov/emp/tables/industry-employment-and-output.htm

[4] https://www.bls.gov/emp/frequently-asked-questions.htm

[5] As one example, Ahern and Harford (2014) investigate the propagation of M&A activities through industry links. Their work is based on the input–output data from the Bureau of Economic Analysis, which are impossible to retrieve without knowing the assignment of firms' NAICS industries.

[6] https://www.census.gov/naics/?input=52&chart=2012&details=524113

A deep grasp of definition-based knowledge allows an industry assignment method to form an informative expectation about which industry a firm should be assigned to, because this type of knowledge defines the scope of business activities covered by an industry. Structure-based knowledge refers to the hierarchical way of organizing industries. The hierarchy is presented as a tree structure (e.g., the NAICS structure illustrated in Figure 1) and reveals how industries are related to each other in terms of the business activities they cover. An industry assignment method could leverage the knowledge of industry relatedness to infer which firms should be classified into an industry based on firms that have already been assigned to its related industries. Assignment-based knowledge refers to prior firm–industry assignments by domain experts. While definition-based knowledge pinpoints the core concepts shaping an industry, assignment-based knowledge instantiates these concepts and reflects the knowledge of domain experts who classify firms according to an ICS.

However, existing industry assignment methods rely solely on assignment-based knowledge to learn their industry assignment models while ignoring definition-based and structure-based knowledge (Kearney and Kornbau 2005, Rodrigues et al. 2012, Wood et al. 2017). Furthermore, these methods adopt a static view of assignment-based knowledge, as if the entire assignment history were created at a single time point, while assignment-based knowledge is actually accumulated incrementally and evolves dynamically over time. Typically, the set of firms assigned to an industry changes over time because new firms might be added to the industry while existing firms might be reassigned to other industries or even removed from the firm universe. Therefore, in addition to definition-based knowledge, the exact business activities covered by an industry also depend on the choices of human experts who keep fine-tuning their interpretation of the industry definition in response to continuous economic innovations. Nevertheless, existing studies only consider which industry a firm has been assigned to, and ignore the time-specificity of assignment-based knowledge, i.e., when the assignment occurs.

It is challenging to design a method that simultaneously considers the three types of expert knowledge discussed above because of the heterogeneity of the data formats: structure-based knowledge is encoded as a tree, definition-based knowledge is presented textually, and assignment-based knowledge is time-stamped. The central challenge is to develop an embedding space where all three types of expert knowledge can be properly represented and integrated for industry assignment. To address the research gaps discussed above, we propose a novel deep learning-based method. In contrast to existing industry assignment methods, which rely solely on assignment-based knowledge, our method seamlessly integrates definition-based, structure-based, and assignment-based knowledge for industry assignment. Moreover, our method considers the time-specificity of assignment-based knowledge, which is neglected by existing industry assignment methods. In doing so, our study

makes two methodological contributions: dynamic industry representation and hierarchical assignment. Dynamic industry representation embeds an industry as a sequence of time-specific vectors by integrating the three types of knowledge through our proposed temporal and spatial aggregation mechanisms. Each vector represents the industry in a specific time period. Thus, dynamic industry representation distinguishes our method from existing industry assignment methods, which treat an industry as a static class label. Hierarchical assignment computes the time-specific probability of assigning a firm to a focal-level industry. It considers industries across the ICS hierarchy and incorporates structure-based knowledge into the probability computation, unlike existing industry assignment methods that only consider industries at the focal level and ignore structure-based knowledge.

2. Literature Review

Two streams of research are closely related to our study: existing methods for industry assignment and state-of-the-art machine and deep learning models that can be adapted to process structure-based or definition-based knowledge. In this section, we review each stream of related research and highlight the key novelties of our study.

2.1. Fintech and Industry Assignment

Our study generally falls into the research field of Fintech (Hendershott et al. 2017, Goldstein et al. 2019). In this field, IS researchers have actively developed advanced IT solutions to solve a diverse set of critical financial problems. For example, Hu et al. (2012) treat firms (banks) that are linked by their financial relationships as a network and develop a network-based method to analyze firms' financial risks. Bao and Datta (2014) also study financial risks, but from a different perspective: they propose a topic model-based method to quantify firms' financial risks based on their textual risk disclosures. Abbasi et al. (2012) focus on financial fraud and propose a meta-learning framework to detect firms' financial frauds. We study the industry assignment problem and contribute to the Fintech field with a novel and effective solution method for this important problem.[7] A research stream related to the industry assignment problem attempts to design new ICSs by clustering firms with similar economical characteristics into homogeneous groups (Jaffe 1986, Lee et al. 2015, Hoberg and Phillips 2016, Gao et al. 2020). For example, Jaffe (1986) represents each firm as a vector of its distribution over patent classes and computes pairwise similarities between firms based on their corresponding vectors; Lee et al. (2015) measure the similarity between two firms based on how frequently they are co-searched on the EDGAR website; and Hoberg and Phillips (2016)

[7]The basic classification unit of an ICS is usually a firm. In some cases, a smaller classification unit might be used, e.g., a single factory among many factories operated by a firm. Our method is still applicable to classify these smaller units if their business descriptions are provided.

represent each firm as a bag-of-words vector derived from its annual 10-K report, and compute the cosine similarity between firm vectors. Different from these studies, our study tackles the industry assignment problem, which assigns firms into industries of an existing ICS and is essentially a classification problem.

Early methods use keyword matching techniques to automate industry assignment (Chen et al. 1993, Kearney and Kornbau 2005, Lim et al. 2005, Jung et al. 2008). For example, Kearney and Kornbau (2005) document a two-step approach that assigns NAICS industries to newly birthed firms based on information collected from their application forms for employer identification numbers. In the first step, the words in an application form are matched against a set of keyword dictionaries, each of which corresponds to an industry and is constructed from historical firm–industry assignments. This step generates a set of candidate industries for a firm. The second step employs a logistic regression to select the most probable industry from the candidates. These methods rely on the quality of the keyword dictionaries, the construction of which is labor-intensive. Thus, they do not scale to ICSs with a large number of industries.

Modern industry assignment methods utilize more sophisticated machine learning models, thereby avoiding the scalability issue (Roelands et al. 2010, Thompson et al. 2012, Rodrigues et al. 2012, Gweon et al. 2017, Wood et al. 2017). These methods formulate the industry assignment problem as a multi-class classification problem and learn a classification model from prior firm–industry assignments conducted by domain experts. The learned model can then be used to classify a new firm into one of the focal-level industries. For instance, Roelands et al. (2010) classify a firm into a top-level industry of the NACE (a European ICS) taxonomy by using texts from the firm's website as inputs. They experiment with several machine learning methods, including support vector machines (SVMs). Deep learning models have also been utilized for industry assignment (Wood et al. 2017, Tagarev et al. 2019, Wei et al. 2019). Wood et al. (2017) apply a multi-layer perceptron model to assign firms to NAICS industries. The model is trained with prior firm–industry assignments, and the input feature for a firm, or firm representation, is a high-dimensional, sparse bag-of-words vector encoding the textual information about the firm. Subsequent studies have investigated more sophisticated text representation methods for firm representation. Specifically, Tagarev et al. (2019) compare four methods of representing firms' description documents as input features for industry assignment, and conclude that the ULMFiT method (Howard and Ruder 2018) achieves the best industry assignment performance. Wei et al. (2019) propose a novel firm representation method by combining both textual and non-textual information about a firm via an attention-based recurrent neural network architecture. Tagarev et al. (2019) and Wei et al. (2019) focus on firm representation but ignore industry representation, and these two studies employ an existing classification method for firm-industry assignment. Our study, on the other hand, develops

novel approaches for industry representation and firm-industry assignment, but uses an existing method, i.e., Doc2Vec (Le and Mikolov 2014), for firm representation. As discussed in Section 1, definition-based, structure-based, and assignment-based knowledge are all essential for effective industry assignment. However, existing industry assignment methods neglect definition-based and structure-based knowledge, which dampens their classification performance. Recent developments in machine and deep learning provide models that can be adapted to process definition-based or structure-based knowledge for industry assignment; we review these models next.

2.2. Hierarchical Classification and Label Embedding

Hierarchical classification methods can be adapted to process structure-based knowledge. These methods use a tree-shaped label hierarchy (e.g., structure-based knowledge in our study) to organize a set of classifiers, each of which is contained in a non-leaf node of the tree and is only responsible for classifying an instance into one of the child classes of the node (Koller and Sahami 1997, McCallum et al. 1998, Dekel et al. 2004, Weinberger and Chapelle 2009, Silla and Freitas 2011).[8] The classifier at a non-leaf node is learned from training instances whose labels belong to the branches originating from the node. When classifying an instance, the root classifier first predicts it as one of the first-level classes. Next, the classifier at the node of the predicted class is invoked to predict the instance's class at the next level. This step is repeated until a leaf-level class is predicted. Alternatively, a tree-shaped label hierarchy can be used to compose vector representations of classes (Dekel et al. 2004, Weinberger and Chapelle 2009). For example, Dekel et al. (2004) propose representing a class as the summation of vectors representing its ascendant classes in the label hierarchy. These representation vectors are learned from training instances and are then employed to predict the class of an unassigned instance.

Label embedding methods are suitable for leveraging definition-based knowledge. In a label embedding method, each label or instance is described by a textual document (i.e., a sequence of words) and represented by a numeric vector that captures the semantics of the document (Yazdani and Henderson 2015, Nam et al. 2016, Pappas and Henderson 2019). Yazdani and Henderson (2015) propose representing a label document by concatenating its component word vectors and encoding an instance document with the summation of its component word vectors, where the word vectors are learned using a pretrained word embedding model. Nam et al. (2016) apply a document embedding model proposed by Le and Mikolov (2014) to encode both label and instance documents. In Pappas and Henderson (2019), a label document is represented as the average of its component word vectors, whereas an instance document is encoded using a sequential compositional neural network. The compatibility score between an instance–label pair is computed by applying

[8] In the machine learning literature, classes in a classification model are more generally referred to as labels.

a matching function to their representation vectors. The matching function can be a simple inner product of label and instance vectors (Yazdani and Henderson 2015) or a more complicated neural network that takes label and instance vectors as inputs (Pappas and Henderson 2019). Finally, the parameters of a label embedding model are learned in such a way that the correct label for a training instance has the highest compatibility score among all of the labels. By treating industry definitions as label documents, we can adapt label embedding methods to process definition-based knowledge.

2.3. Key Novelties of Our Method

Our literature review suggests several research gaps. Existing industry assignment methods only consider assignment-based knowledge but ignore the time specificity of this knowledge; they also neglect definition-based and structure-based knowledge. While hierarchical classification methods can be adapted to process structure-based knowledge, these methods are not designed to leverage definition-based knowledge. On the other hand, label embedding methods are suitable for processing definition-based knowledge, but they do not work with structure-based knowledge. Moreover, both hierarchical classification and label embedding methods neglect the time specificity of assignment-based knowledge. Our method simultaneously considers the three types of knowledge as well as the time specificity of assignment-based knowledge. Therefore, our method is distinct from existing methods, as summarized in Table 1. Methodologically, our method features two innovations: dynamic industry representation and hierarchical assignment. Dynamic industry representation embeds an industry as a sequence of time-specific vectors, in contrast to existing industry assignment methods and hierarchical classification methods that represent an industry as a static class label. It derives these time-specific vectors by leveraging the three types of knowledge. Label embedding methods, in contrast, represent an industry as a static vector based solely on definition-based knowledge. Furthermore, hierarchical assignment distinguishes our method from existing methods in that it incorporates structure-based knowledge into the computation of the time-specific probability of assigning a firm to an industry.

3. Problem Formulation

We consider a target ICS in which the industries are organized hierarchically as an "industry tree". Figure 2 shows the industry tree for an example ICS that will be used throughout this paper. In general, the ith industry at level l is denoted as $\mathcal{T}_{li}$, assuming an arbitrary ordering of the industries at the level. Let N_l be the number of industries at level l. At the root level, $l = 0$ and $N_0 = 1$, which means that $\mathcal{T}_{01}$ is the single root node that represents the entire ICS. At the leaf level, $l = L$, which

Table 1 Comparison between Our Method and Existing Methods in Terms of the Used Knowledge

	Definition-based Knowledge	Structure-based Knowledge	Assignment-based Knowledge*
Existing Industry Assignment Methods, e.g., Wood et al. (2017), Tagarev et al. (2019)	No	No	Static
Hierarchical Classification Methods, e.g., Silla and Freitas (2011)	No	Yes	Static
Label Embedding Methods, e.g., Pappas and Henderson (2019)	Yes	No	Static
Our Method	Yes	Yes	Dynamic

* For the industry assignment problem, an instance of firm-industry assignment consists of three pieces of information: a firm, the industry the firm was assigned to, and the timestamp when the assignment occurred. An industry assignment method is based on static assignment-based knowledge if it only considers the assigned firm-industry pair but ignores the time-specificity of the assignment, while it is based on dynamic assignment-based knowledge if it utilizes all three pieces of information.

indicates that the industry tree contains L levels in total without counting the root level. The set of industries at level l is written as

$$\mathcal{T}_l = \{\mathcal{T}_{l1}, \mathcal{T}_{l2}, \ldots, \mathcal{T}_{lN_l}\},$$

and the complete set of industries is $\mathcal{T} = \cup_{l=1}^{L} \mathcal{T}_l$. The total number of industries is computed as $N = \sum_{l=1}^{L} N_l$, excluding the root node. The target ICS is applied to a firm universe $\mathcal{U}$.

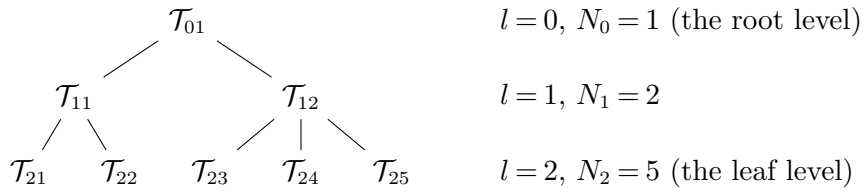

Figure 2 Industry Tree for an Example ICS ($N = 7$, $L = 2$, excluding $\mathcal{T}_{01}$)

Given a focal level of interest l^*, $1 \leq l^* \leq L$, there exists a set of firm–industry assignment cases that are accumulated in periods $1:T$, which is short for $[1, 2, \ldots, T]$. Each assignment case is a tuple $(j, y^{(t,j)})$ comprised of a firm and its industry assigned in a particular period, where $j \in \mathcal{U}$ represents a firm and $y^{(t,j)}$ is the industry at level l^* assigned to firm j in period $t \in 1:T$. Let $doc^{(t,j)}$ denote the document describing the business of firm j in period t. We can now formally define the industry assignment problem:

Definition 1 (**Industry Assignment Problem**). Given a target ICS in which the industries $\mathcal{T}$ are organized as a tree, a corpus of textual definition for each industry in $\mathcal{T}$, a firm universe $\mathcal{U}$, its associated corpus $\{doc^{(t,j)} \mid j \in \mathcal{U},\ t \in 1:T\}$ of firms' business description documents from period 1 to T, a focal industry level l^*, and a set of past firm–industry assignment cases performed by domain experts

$$\mathcal{D} = \{(j, y^{(t,j)}) \mid j \in \mathcal{U},\ y^{(t,j)} \in \mathcal{T}_{l^*},\ t \in 1:T\}, \tag{1}$$

build a model to classify each unassigned firm $j' \in \mathcal{U}$ in period $T+1$ to an industry in $\mathcal{T}_{l^*}$ based on the business description document $doc^{(T+1,j')}$ of the firm in period $T+1$, such that the assigned industry best covers the business activities of the firm, which can be evaluated by comparing the industry $\hat{y}^{(T+1,j')}$ assigned by the model and the ground truth industry $y^{(T+1,j')}$ assigned by domain experts over all unassigned firms according to some metrics of interest.

4. DeepIA: A Deep Learning Method for Industry Assignment

We present the three building blocks of DeepIA in Sections 4.1, 4.2, and 4.3 and then introduce DeepIA in Section 4.4. The first building block encodes the three types of expert knowledge and represents firms; the second represents industries based on the encoded knowledge; and the third takes firm and industry representations as inputs and computes the probability of assigning a firm to an industry. The last two building blocks constitute main methodological novelties of our study and we highlight these novelties at the end of Sections 4.2 and 4.3. For the convenience of the reader, we summarize important notation in Table 2.

Table 2 Notation

Notation	Description
$\mathcal{T}_{li}$	The ith industry at level l
$\mathcal{T}_{li}^{(D)}$	Definition-based knowledge of $\mathcal{T}_{li}$, Equation 2
$\mathcal{T}_{li}^{(S)}$	Structure-based knowledge of $\mathcal{T}_{li}$, Equation 3
$\mathcal{T}_{li}^{(A,t)}$	Assignment-based knowledge of $\mathcal{T}_{li}$ in period t, Equation 4
$\mathcal{P}$	Ancestor seeking operator, Definition 2
$\mathcal{C}$	Descendant seeking operator, Definition 2
$x^{(t,j)}$	Representation of firm j in period t, Equation 5
$v_{li}^{(t)}$	Dynamic representation of industry $\mathcal{T}_{li}$ in period t, Equation 17
$P(\mathcal{T}_{li}\|j,t)$	Probability that firm j is assigned to industry $\mathcal{T}_{li}$ in period t among all the industries at level l, Equation 23

4.1. Encoding Expert Knowledge and Representing Firms

We use $\mathcal{T}_{li}^{(D)}$, $\mathcal{T}_{li}^{(S)}$, and $\mathcal{T}_{li}^{(A,t)}$, respectively, to denote the definition-based, structure-based, and assignment-based knowledge of industry $\mathcal{T}_{li}$. It is straightforward to store the paragraph defining an industry as a sequence of words

$$\mathcal{T}_{li}^{(D)} =< w_1^{(li)}, w_2^{(li)}, \ldots, w_k^{(li)}, \cdots >, \tag{2}$$

where $w_k^{(li)}$ is the kth word in the paragraph defining industry $\mathcal{T}_{li}$. To express structure-based knowledge, we define two operators that apply to the industry tree of an ICS:

Definition 2 (**Ancestor- and Descendant-Seeking Operators**). Let $\mathcal{P}$ be the ancestor-seeking operator such that $\mathcal{P}^k(\mathcal{T}_{li})$ returns the unique ancestor industry of $\mathcal{T}_{li}$ at level $l-k$ (i.e., k levels above level l) for $1 \le l \le L$ and $1 \le k \le l$, and $\mathcal{P}^0(\mathcal{T}_{li})$ returns $\mathcal{T}_{li}$ itself. Similarly, let $\mathcal{C}$ be the descendant-seeking operator such that $\mathcal{C}^k(\mathcal{T}_{li})$ returns the set of descendant industries of $\mathcal{T}_{li}$ at level $l+k$ (i.e., k levels below level l) for $0 \le l \le L-1$ and $1 \le k \le L-l$, and $\mathcal{C}^0(\mathcal{T}_{li})$ returns $\mathcal{T}_{li}$ itself.

For simplicity, $\mathcal{P}(\mathcal{T}_{li})$ means $\mathcal{P}^1(\mathcal{T}_{li})$, or the parent industry of $\mathcal{T}_{li}$, and $\mathcal{C}(\mathcal{T}_{li})$ means $\mathcal{C}^1(\mathcal{T}_{li})$, or the set of child industries of $\mathcal{T}_{li}$. Using the defined operators, the structure-based knowledge of industry $\mathcal{T}_{li}$ is comprised of all of its ancestor and descendant industries (excluding the root node):

$$\mathcal{T}_{li}^{(S)} = \begin{cases} \{\mathcal{C}(\mathcal{T}_{li}), \mathcal{C}^2(\mathcal{T}_{li}), \ldots, \mathcal{C}^{L-l}(\mathcal{T}_{li})\} & \text{if } l = 1 \\ \{\mathcal{P}(\mathcal{T}_{li}), \mathcal{P}^2(\mathcal{T}_{li}), \ldots, \mathcal{P}^{l-1}(\mathcal{T}_{li}), \mathcal{C}(\mathcal{T}_{li}), \mathcal{C}^2(\mathcal{T}_{li}), \ldots, \mathcal{C}^{L-l}(\mathcal{T}_{li})\} & \text{if } 2 \le l \le L-1 \\ \{\mathcal{P}(\mathcal{T}_{li}), \mathcal{P}^2(\mathcal{T}_{li}), \ldots, \mathcal{P}^{l-1}(\mathcal{T}_{li})\} & \text{if } l = L \end{cases} \tag{3}$$

Example 1. Consider the target ICS in Figure 2. We have $\mathcal{P}(\mathcal{T}_{25}) = \{\mathcal{T}_{12}\}$ and $\mathcal{C}(\mathcal{T}_{12}) = \{\mathcal{T}_{23}, \mathcal{T}_{24}, \mathcal{T}_{25}\}$. By Equation 3, $\mathcal{T}_{12}^{(S)} = \{\mathcal{C}(\mathcal{T}_{12})\} = \{\mathcal{T}_{23}, \mathcal{T}_{24}, \mathcal{T}_{25}\}$ and $\mathcal{T}_{25}^{(S)} = \{\mathcal{P}(\mathcal{T}_{25})\} = \{\mathcal{T}_{12}\}$.

The assignment-based knowledge of industry $\mathcal{T}_{li}$ in period t, denoted as $\mathcal{T}_{li}^{(A,t)}$, is the set of firms assigned to the industry in that period. Given past firm–industry assignments $\mathcal{D}$, we know which firm is assigned to which industry at focal level l^* in which time period. Note that a firm is assigned to an industry if it is assigned to one of the industry's descendants at level l^*. Therefore, $\mathcal{T}_{li}^{(A,t)}$ consists of firms whose assigned industry in period t belongs to $\mathcal{T}_{li}$'s descendants at level l^*. Formally, it is given by

$$\mathcal{T}_{li}^{(A,t)} = \{j \mid (j, y^{(t,j)}) \in \mathcal{D}, y^{(t,j)} \in \mathcal{C}^{l^*-l}(\mathcal{T}_{li})\}. \tag{4}$$

By Definition 2, $\mathcal{C}^{l^*-l}(\mathcal{T}_{li})$ returns the set of $\mathcal{T}_{li}$'s descendant industries at focal level l^* if $l < l^*$ and $\mathcal{C}^{l^*-l}(\mathcal{T}_{li}) = \mathcal{C}^0(\mathcal{T}_{li}) = \{\mathcal{T}_{li}\}$ if $l = l^*$. Note that if $l > l^*$, firms assigned to industry $\mathcal{T}_{li}$ cannot be inferred from $\mathcal{D}$. Hence, we can only obtain assignment-based knowledge for industries at level l for $l \le l^*$.

Example 2. Consider the target ICS in Figure 2 and focal level $l^* = 2$. Past firm–industry assignments $\mathcal{D}$ in time periods $t = 1, 2$ are as follows:

$$\mathcal{D} = \{(B, y^{(1,B)} = \mathcal{T}_{21}), (C, y^{(1,C)} = \mathcal{T}_{23}), (D, y^{(1,D)} = \mathcal{T}_{25}), (E, y^{(1,E)} = \mathcal{T}_{22}),$$
$$(B, y^{(2,B)} = \mathcal{T}_{21}), (C, y^{(2,C)} = \mathcal{T}_{25}), (D, y^{(2,D)} = \mathcal{T}_{25}), (F, y^{(2,F)} = \mathcal{T}_{24})\}.$$

In $\mathcal{D}$, each uppercase letter represents a firm, and a tuple such as $(B, y^{(1,B)} = \mathcal{T}_{21})$ indicates that firm B is assigned to industry $\mathcal{T}_{21}$ in period 1. According to $\mathcal{D}$, firms B, C, D, E are assigned to industries $\mathcal{T}_{21}, \mathcal{T}_{23}, \mathcal{T}_{25}, \mathcal{T}_{22}$, respectively, in period 1. In period 2, firms B and D stay in the same industry as period 1, while firm C switches from $\mathcal{T}_{23}$ to $\mathcal{T}_{25}$. In addition, firm E is removed from the firm universe in period 2, while firm F is newly added to the universe in period 2 and is assigned to industry $\mathcal{T}_{24}$. Figure 3 illustrates the assignment-based knowledge derived from $\mathcal{D}$. As shown,

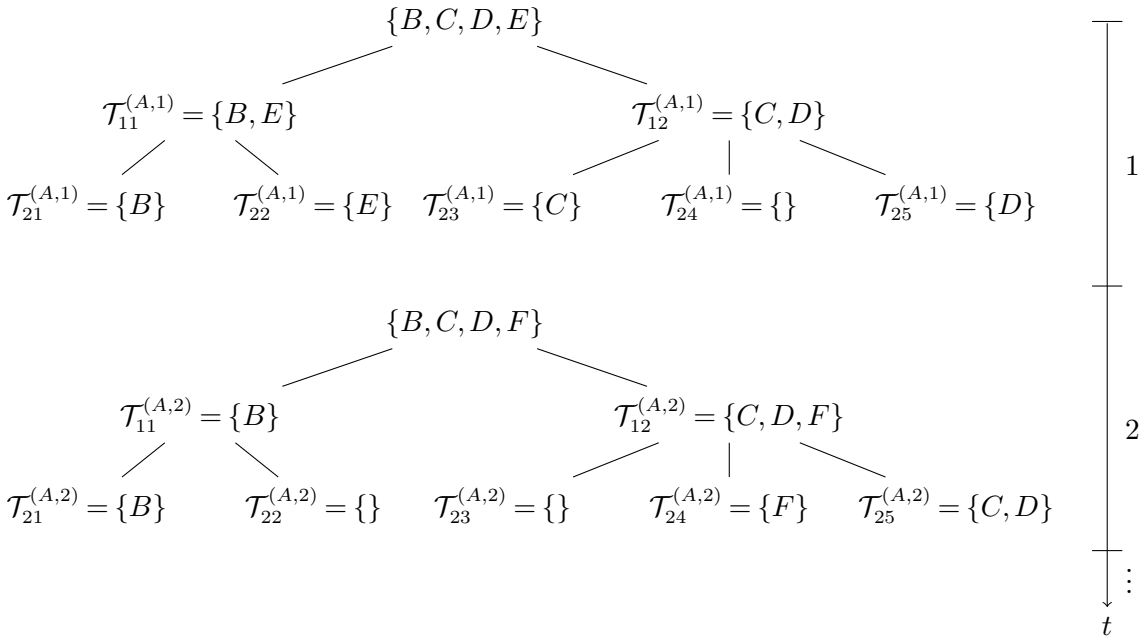

Figure 3 **Assignment-based Knowledge over Two Time Periods**

the assignment-based knowledge $\mathcal{T}_{21}^{(A,1)}$ of industry $\mathcal{T}_{21}$ in period 1 consists of firm B assigned to the industry in that period, i.e., $\mathcal{T}_{21}^{(A,1)} = \{B\}$. Similarly, we have $\mathcal{T}_{22}^{(A,1)} = \{E\}$. By Equation 4, the assignment-based knowledge $\mathcal{T}_{11}^{(A,1)}$ of industry $\mathcal{T}_{11}$ in period 1 contains firms assigned to its descendant industries (i.e., $\mathcal{T}_{21}$ and $\mathcal{T}_{22}$) in that period, and hence $\mathcal{T}_{11}^{(A,1)} = \{B, E\}$. Assignment-based knowledge is time-specific, thereby capturing the dynamics of industry assignment. For instance, the assignment-based knowledge of industry $\mathcal{T}_{11}$ changes from $\{B, E\}$ in period 1 to $\{B\}$ in period 2, as firm E is removed from the firm universe in period 2.

We embed firm j in period t as a d-dimensional vector $x^{(t,j)} \in R^d$. The representation of firm j is time-specific because it should reflect the most up-to-date business activities of the firm. We derive $x^{(t,j)}$ from $doc^{(t,j)}$, the business description document for firm j in period t.[9] To this end, we employ a document embedding model (DEM), e.g., the Doc2Vec model developed by Le and Mikolov (2014), which summarizes the semantics of each input document as a numeric vector. Specifically, we have

$$x^{(t,j)} = e_f\big(\text{DEM}(doc^{(t,j)})\big), \tag{5}$$

where DEM denotes a document embedding model and e_f is a transformation layer. Because business description documents are typically long, it is computationally challenging to learn the parameters for a DEM as part of the model. Therefore, following Pappas and Henderson (2019), the DEM in Equation 5 is pretrained. Because the DEM is trained without using the firm-industry assignment information, the resulted document vector might contain information that is irrelevant for industry assignment. To filter out the irrelevant information, we employ a transformation layer e_f (e.g., a multi-layer perceptron) to extract informative firm representation $x^{(t,j)}$ from $\text{DEM}(doc^{(t,j)})$, where "informative" means "informative for industry assignment". The parameters for e_f are learned as part of the model parameters. The choices of the DEM and e_f depend on the characteristics of the input documents (e.g., document length), and hence we specify them when reporting our experiments in Section 5.

4.2. Dynamic Industry Representation

The central idea of dynamic industry representation (DIR) is to represent an industry $\mathcal{T}_{li}$ as a sequence of time-specific vectors $\langle v_{li}^{(t)} \rangle$, $t = 1, 2, \ldots, T$, each of which coherently integrates $\mathcal{T}_{li}$'s definition-based knowledge $\mathcal{T}_{li}^{(D)}$, structure-based knowledge $\mathcal{T}_{li}^{(S)}$, and assignment-based knowledge up to time period t: $\mathcal{T}_{li}^{(A,1)}, \mathcal{T}_{li}^{(A,2)}, \ldots, \mathcal{T}_{li}^{(A,t)}$. To achieve this goal, $\mathcal{T}_{li}^{(D)}$ and $\mathcal{T}_{li}^{(A,t)}$ are respectively embedded as vectors $v_{li}^{(D)} \in R^d$ and $v_{li}^{(A,t)} \in R^d$, $t = 1, 2, \ldots, T$. Next, embedding vectors $v_{li}^{(D)}$, $v_{li}^{(A,1)}, v_{li}^{(A,2)}, \ldots, v_{li}^{(A,t)}$ are integrated to form a DIR vector $v_{li}^{(t)}$. In Section 4.2.1, we propose a spatial aggregation mechanism that derives $v_{li}^{(D)}$ and $v_{li}^{(A,t)}$ in a bottom-up fashion to incrementally fuse information from lower-level industries to upper-level industries. Then, in Section 4.2.2, we present a temporal aggregation mechanism that combines these embedding vectors to form a DIR vector. We summarize DIR and its methodological novelties in Section 4.2.3.

[9] A firm's business description document might come from different sources. It can be extracted from the firm's website (Wood et al. 2017) or the firm's annual financial report (i.e., a 10-K report) (Pierre 2001). In general, such a document describes the business activities in which the firm engages and is therefore an ideal textual source for industry assignment.

4.2.1. Spatial Aggregation We propose a spatial aggregation mechanism to represent the assignment-based knowledge about industry $\mathcal{T}_{li}$ in period t as a vector $v_{li}^{(A,t)} \in R^d$ and embed its definition-based knowledge as a vector $v_{li}^{(D)} \in R^d$. We begin by embedding the assignment-based knowledge.

At the leaf level (i.e., $l = L$), $v_{Li}^{(A,t)}$ is randomly initialized and learned as part of the model parameters. However, at level $1 \leq l \leq L-1$, $v_{li}^{(A,t)}$ is derived from the child industries of $\mathcal{T}_{li}$:

$$v_{li}^{(A,t)} = \sum_{k \in \mathcal{C}(\mathcal{T}_{li})} \beta_k^{(li,t)} v_{(l+1)k}^{(A,t)}, \tag{6}$$

where $0 \leq \beta_k^{(li,t)} \leq 1$, $\sum_{k \in \mathcal{C}(\mathcal{T}_{li})} \beta_k^{(li,t)} = 1$, $k \in \mathcal{C}(\mathcal{T}_{li})$ is short for $\mathcal{T}_{(l+1)k} \in \mathcal{C}(\mathcal{T}_{li})$, indicating that $\mathcal{T}_{(l+1)k}$ is a child industry of $\mathcal{T}_{li}$ (similar notation is adopted elsewhere), and $v_{(l+1)k}^{(A,t)}$ denotes the representation of $\mathcal{T}_{(l+1)k}$'s assignment-based knowledge in period t. The rationale behind Equation 6 is that the firms assigned to $\mathcal{T}_{li}$ are the collection of firms that are assigned to its child industries. Hence, in the equation, we represent $\mathcal{T}_{li}$'s assignment-based knowledge as a convex combination of the representations of its child industries' assignment-based knowledge. Clearly, the equation synthesizes assignment-based and structure-based knowledge. It also captures the intuition that child industries are of varying importance through $\beta_k^{(li,t)}$. For example, a child industry abundant in assignment cases might enjoy a larger weight than a child industry with few assignment cases. Equation 6 is applied recursively to levels $l = L-1, L-2, \ldots, 1$ to represent assignment-based knowledge for every industry at each level.

Equation 6 is implemented based on the multi-head self-attention framework (Vaswani et al. 2017). Since the following computations are generic and applicable to any industry in any time period, we simplify the notation by rewriting $v_{(l+1)k}^{(A,t)}$ as h_k, dropping the superscript of $\beta_k^{(li,t)}$, and using $\sum_k$ to mean $\sum_{k \in \mathcal{C}(\mathcal{T}_{li})}$. As a result, Equation 6 becomes

$$v_{li}^{(A,t)} = \sum_k \beta_k h_k. \tag{7}$$

Next, for expository purposes, we focus on the case of one attention head. As the first step, we define

$$\tilde{v}_{li}^{(A,t)} = W^{(O)} \left(\sum_k \beta_k W^{(V)} h_k \right), \tag{8}$$

where $W^{(V)} \in R^{d \times d}$ is the matrix that projects each h_k to a value vector and $W^{(O)} \in R^{d \times d}$ projects the summation to the output vector $\tilde{v}_{li}^{(A,t)} \in R^d$.[10] The attention weight β_k should measure the

[10] When there is more than one attention head, each attention head produces a single vector $\sum_k \beta_k W^{(V)} h_k$ specific to that attention head. These vectors are then concatenated into one vector, the length of which equals hd, where h is the number of attention heads. In this case, $W^{(O)} \in R^{d \times hd}$ is introduced to project the concatenated vector to the output vector $\tilde{v}_{li}^{(A,t)} \in R^d$.

relevance of the assignment-based knowledge of the child industry $\mathcal{T}_{(l+1)k}$ to its parent industry $\mathcal{T}_{li}$. To this end, we need a representation of industry $\mathcal{T}_{li}$ that serves as the query vector. We do this by reusing the embedding vectors $\{v^{(A,t)}_{(l+1)k} |\ k \in \mathcal{C}(\mathcal{T}_{li})\}$, or $\{h_k\}$ for simplicity, which are available at the time of computing $v^{(A,t)}_{li}$. This design avoids introducing additional parameters for the query vector. Let Z be the matrix formed by stacking the vectors $\{h_k\}$ column-wise. Let g be a function that transforms an input matrix to a vector by taking the mean of each row of the input matrix.[11] Then we use $g(Z)$ as the query representation of $\mathcal{T}_{li}$ up to a linear transformation, and define β_k as

$$\beta_k = \frac{\exp\Big(\big(W^{(Q)} g(Z)\big)^T (W^{(K)} h_k)/\sqrt{d}\Big)}{\sum_{k'} \exp\Big(\big(W^{(Q)} g(Z)\big)^T (W^{(K)} h_{k'})/\sqrt{d}\Big)}, \tag{9}$$

where $W^{(Q)} \in R^{d\times d}$ projects $g(Z)$ to a query vector and $W^{(K)} \in R^{d\times d}$ projects each h_k to a key vector. [12]Equation 9 states that β_k is measured as the relevance of the piece of knowledge h_k to $g(Z)$, the query representation of parent industry $\mathcal{T}_{li}$, up to a linear transformation. Finally, to obtain $v^{(A,t)}_{li}$, a post-transformation layer is added on top of $\tilde{v}^{(A,t)}_{li}$ to introduce non-linearity:

$$v^{(A,t)}_{li} = \max\big(W_1(g(Z) + \tilde{v}^{(A,t)}_{li}) + b_1, 0\big), \tag{10}$$

where $W_1 \in R^{d\times d}$ and $b_1 \in R^d$. Equation 10 adds $g(Z)$, the query representation of $\mathcal{T}_{li}$, to $\tilde{v}^{(A,t)}_{li}$. This design mimics the residual structure of which the purpose is to facilitate the learning of deep neural networks (He et al. 2016). The function max() compares each element of its left argument with zero and returns the greater value. This function is formally called ReLU and is used to introduce non-linearities (Goodfellow et al. 2016).

The computation flow from Equations 8 to 10 is applicable to any industry in any time period and can be generalized as a function $MHA(q, K, V|\Theta)$. The objective of the function is to summarize n pieces of knowledge (e.g., n equals the size of $\mathcal{C}(\mathcal{T}_{li})$ in our case) according to a given query q. By applying a linear transformation to q, the function produces a query vector. Similarly, by applying linear transformations to the kth column vectors of K and V, respectively, it generates the key and value vectors for the kth piece of knowledge. The query vector is matched against the set of key vectors (e.g., Equation 9), and the normalized matching scores (e.g., β_k) are used to combine the value vectors as an output vector (e.g., Equation 8). Lastly, the query q is combined with the output vector, and the summation is transformed non-linearly through Equation 10. In the case of spatial aggregation for the assignment-based knowledge of industry $\mathcal{T}_{li}$, we have

$$v^{(A,t)}_{li} = MHA(q = g(Z),\ K = Z,\ V = Z|\Theta), \tag{11}$$

[11]The choice of function g is flexible as long as it aggregates the vectors of a matrix as one vector. Empirically, we have found that the mean function works well.

[12]The term $\sqrt{d}$ in Equation 9 is used to smooth the distribution of attention weights (Vaswani et al. 2017).

where the parameters Θ are defined as

$$\Theta = \{W^{(Q)}, W^{(K)}, W^{(V)}, W_1, b_1, W^{(O)}\}. \tag{12}$$

We have two remarks about the spatial aggregation mechanism defining $v_{li}^{(A,t)}$. First, only the leaf level adds a set of learnable model parameters $\{v_{Li}^{(A,t)} | i = 1, 2, \ldots, N_L\}$. The embedding vectors for the assignment-based knowledge at the upper levels are derived recursively from this set of model parameters through Equation 11. This design avoids adding a huge set of model parameters for industries beyond the leaf level. Second, the set of assignment-based knowledge $\{\mathcal{T}_{li}^{(A,t)} | \mathcal{T}_{li} \in \mathcal{T}\}$ defined by Equation 4 is not used to directly derive the knowledge embedding vectors $\{v_{li}^{(A,t)} | \mathcal{T}_{li} \in \mathcal{T}\}$, but is rather used to indirectly shape them through the learning objective formulated in Section 4.4.

To embed the definition-based knowledge $\mathcal{T}_{li}^{(D)}$, we employ a DEM that summarizes the semantics of $\mathcal{T}_{li}^{(D)}$ as a vector $\bar{v}_{li}^{(D)} \in R^d$, i.e.,

$$\bar{v}_{li}^{(D)} = \text{DEM}(\mathcal{T}_{li}^{(D)}). \tag{13}$$

The choice of DEM is specified in Section 5. Similar to embedding assignment-based knowledge, at the leaf level we set $v_{Li}^{(D)} = \bar{v}_{Li}^{(D)}$, while at level $1 \le l \le L-1$ we derive $v_{li}^{(D)}$ from $\bar{v}_{li}^{(D)}$ as well as the child industries of $\mathcal{T}_{li}$:

$$v_{li}^{(D)} = \gamma_0^{(li)} \bar{v}_{li}^{(D)} + \sum_{k \in \mathcal{C}(\mathcal{T}_{li})} \gamma_k^{(li)} v_{(l+1)k}^{(D)}, \tag{14}$$

where $0 \le \gamma_0^{(li)} \le 1$, $0 \le \gamma_k^{(li)} \le 1$, $\gamma_0^{(li)} + \sum_{k \in \mathcal{C}(\mathcal{T}_{li})} \gamma_k^{(li)} = 1$, $k \in \mathcal{C}(\mathcal{T}_{li})$ indicates that $\mathcal{T}_{(l+1)k}$ is a child industry of $\mathcal{T}_{li}$, and $v_{(l+1)k}^{(D)}$ denotes the representation of $\mathcal{T}_{(l+1)k}$'s definition-based knowledge. The intuition of Equation 14 is that business activities covered by any of $\mathcal{T}_{li}$'s child industries should also be covered by $\mathcal{T}_{li}$. Hence, $v_{li}^{(D)}$ depends on not only the embedded definition-based knowledge of $\mathcal{T}_{li}$ but also the definition-based knowledge of its child industries. To implement Equation 14, let Y be the matrix formed by stacking the vectors $\{\bar{v}_{li}^{(D)}\} \cup \{v_{(l+1)k}^{(D)} | k \in \mathcal{C}(\mathcal{T}_{li})\}$ column-wise. The definition-based knowledge for $1 \le l \le L-1$ is then embedded as

$$v_{li}^{(D)} = MHA(q = g(Y),\ K = Y,\ V = Y | \Theta), \tag{15}$$

where the spatial aggregation parameters Θ are given in Equation 12.

4.2.2. Temporal Aggregation We formulate the DIR vector $v_{li}^{(t)}$ for an industry $\mathcal{T}_{li}$ as a convex combination of vectors $v_{li}^{(D)}, v_{li}^{(A,1)}, v_{li}^{(A,2)}, \ldots, v_{li}^{(A,t)}$, which represent $\mathcal{T}_{li}$'s definition-based knowledge and assignment-based knowledge up to period t. Accordingly, we have

$$v_{li}^{(t)} = \alpha_0^{(li,t)} v_{li}^{(D)} + \alpha_1^{(li,t)} v_{li}^{(A,1)} + \alpha_2^{(li,t)} v_{li}^{(A,2)} + \cdots + \alpha_t^{(li,t)} v_{li}^{(A,t)}, \tag{16}$$

where $0 \leq \alpha_k^{(li,t)} \leq 1$ and $\sum_{k=0}^{t} \alpha_k^{(li,t)} = 1$. The scalar $\alpha_k^{(li,t)}$ serves as the attention weight placed on the kth piece of knowledge and is specific to industry $\mathcal{T}_{li}$ in period t. Intuitively, if a piece of knowledge is useful for classifying a firm, it should have a relatively large weight. Attention weights vary across industries and time for the following reasons. First, some industries might be more informatively defined than others by using more specific words conveying their covered business activities. As a result, definition-based knowledge of different industries is not equally important for industry assignment. Second, attention weights are time-specific, because the importance of pieces of knowledge should be evaluated within the knowledge set that is available by the time period considered and reevaluated when new pieces of knowledge are introduced in subsequent periods.

We implement Equation 16 in a similar way to the spatial aggregation mechanism. Using the notation $MHA(q, K, V)$ developed in Section 4.2.1, we define

$$v_{li}^{(t)} = MHA(q = v_{li}^{(A,t)},\ K = M,\ V = M \mid \Theta'), \tag{17}$$

where M is the matrix formed by stacking the vectors $v_{li}^{(D)}, v_{li}^{(A,1)}, v_{li}^{(A,2)}, \ldots, v_{li}^{(A,t)}$ column-wise. We use $v_{li}^{(A,t)}$ in Equation 17 as the query vector. Because the assignment-based knowledge in period t represents the most up-to-date expert knowledge about which firms should be classified into industry $\mathcal{T}_{li}$, measuring the attention weights based on this knowledge generates an integrated knowledge representation that is best suited for classifying a firm in period t. The computation behind Equation 17 contains two steps. First, a vector $\tilde{v}_{li}^{(t)}$ is computed in a similar way to Equation 8 by averaging knowledge embedding vectors $v_{li}^{(D)}, v_{li}^{(A,1)}, v_{li}^{(A,2)}, \ldots, v_{li}^{(A,t)}$ (which are column vectors of M) based on attention weights for query $v_{li}^{(A,t)}$. Second, a post-transformation layer analogous to Equation 10 is imposed, which is defined as $v_{li}^{(t)} = \max\left(W_1'(v_{li}^{(A,t)} + \tilde{v}_{li}^{(t)}) + b_1', 0\right)$ where $v_{li}^{(A,t)}$ is added to $\tilde{v}_{li}^{(t)}$ to facilitate the learning of our model.

The set of *MHA* parameters in this case is denoted by Θ' and has a similar structure to Θ (Equation 12), but is parameterized independently. Formally, Θ' is given by

$$\Theta' = \{W'^{(Q)}, W'^{(K)}, W'^{(V)}, W_1', b_1', W'^{(O)}\}, \tag{18}$$

and these parameters are shared across industries and time periods.

4.2.3. Summary Dynamic industry representation distinguishes our method from existing industry assignment methods through its novel representation of an industry as a sequence of time-specific vectors that are derived by integrating definition-based, assignment-based, and structure-based knowledge through the proposed temporal and spatial aggregation mechanisms. Although these mechanisms are built upon the multi-head self-attention framework, applying the framework

to our problem requires defining and solving problem-specific details. In this regard, the novelty of the temporal and spatial aggregation mechanisms lies in the formulation of the three embedding vectors in Equations 16, 6, and 14, as well as the specification of queries, keys, and values in Equations 17, 11, 15, which are designed to leverage the three types of expert knowledge for dynamic industry representation.

4.3. Hierarchical Assignment

Given firm representation $x^{(t,j)}$ (Equation 5) and dynamic industry representation $v_{li}^{(t)}$ (Equation 16), we define the compatibility score between firm j and industry $\mathcal{T}_{li}$ in period t as

$$s(\mathcal{T}_{li}, j, t) = \exp({v_{li}^{(t)}}^T x^{(t,j)}), \tag{19}$$

which is the exponential of the inner product of the firm representation and the industry representation. In deep learning, it is common to measure the compatibility between two vector representations using their inner product followed by an exponential transformation (Goodfellow et al. 2016). In our context, the larger the compatibility score, the better the business activities of the firm fit into the scope of the industry.

Let $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t)$ be the probability that in period t, firm j is assigned to industry $\mathcal{T}_{li}$ among all the industries at level l, given that it has already been assigned to the parent industry $\mathcal{P}(\mathcal{T}_{li})$ of $\mathcal{T}_{li}$. To measure $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t)$, we observe the following hierarchy constraint for ICSs:

Definition 3 (**The Hierarchy Constraint**). If a firm is assigned to an industry z, it must also belong to one of industry z's child industries. Likewise, if a firm is assigned to an industry y, it should also belong to y's parent industry.

Consider the ICS in Figure 2 as an example. By the hierarchy constraint, a firm assigned to industry $\mathcal{T}_{11}$ must also belong to either $\mathcal{T}_{21}$ or $\mathcal{T}_{22}$. Similarly, a firm assigned to industry $\mathcal{T}_{23}$ should also belong to $\mathcal{T}_{12}$.

By the hierarchy constraint, if a firm has been assigned to industry $\mathcal{P}(\mathcal{T}_{li})$ (i.e., the parent industry of $\mathcal{T}_{li}$) at level $l+1$, it must belong to one of the child industries of $\mathcal{P}(\mathcal{T}_{li})$ at level l, i.e., $\mathcal{C}(\mathcal{P}(\mathcal{T}_{li}))$. Therefore, given that a firm has been assigned to industry $\mathcal{P}(\mathcal{T}_{li})$ at level $l+1$, the search space for the firm's industry assignment at level l reduces from all of the industries at that level to $\mathcal{C}(\mathcal{P}(\mathcal{T}_{li}))$. Accordingly, we can formulate $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t)$ as the compatibility score $s(\mathcal{T}_{li}, j, t)$ normalized within $\mathcal{C}(\mathcal{P}(\mathcal{T}_{li}))$:

$$P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t) = \frac{s(\mathcal{T}_{li}, j, t)}{\sum_{i' \in \mathcal{C}\left(\mathcal{P}(\mathcal{T}_{li})\right)} s(\mathcal{T}_{li'}, j, t)}, \tag{20}$$

where $s(\mathcal{T}_{li}, j, t)$ can be computed using Equation 19. We illustrate the computation of $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t)$ with the following example.

Example 3. Figure 4 gives an example of compatibility scores between firm j and each industry in the ICS illustrated in Figure 2. Figure 5 shows the probabilities $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}),j,t)$ computed from these compatibility scores using Equation 20. Since $\mathcal{T}_{01}$ denotes the entire ICS and firm j must belong to an industry in the ICS, we have $P(\mathcal{T}_{01}|j,t)=1$. Consider the computation of $P(\mathcal{T}_{11}|\mathcal{T}_{01},j,t)$. Examining the ICS given in Figure 2, we note that $\mathcal{C}(\mathcal{T}_{01})=\{\mathcal{T}_{11},\mathcal{T}_{12}\}$ and $\mathcal{P}(\mathcal{T}_{11})=\{\mathcal{T}_{01}\}$. By Equation 20, we have

$$P(\mathcal{T}_{11}|\mathcal{T}_{01},j,t)=\frac{s(\mathcal{T}_{11},j,t)}{\sum_{i'\in\mathcal{C}(\mathcal{T}_{01})}s(\mathcal{T}_{1i'},j,t)}=\frac{s(\mathcal{T}_{11},j,t)}{s(\mathcal{T}_{11},j,t)+s(\mathcal{T}_{12},j,t)}=0.4.$$

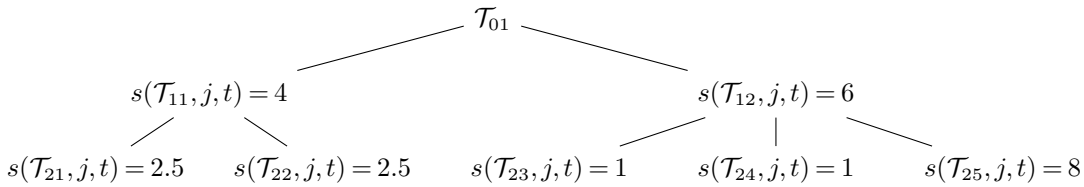

Figure 4 **Compatibility Scores between Firm j and Each Industry in the Example ICS in Figure 2**

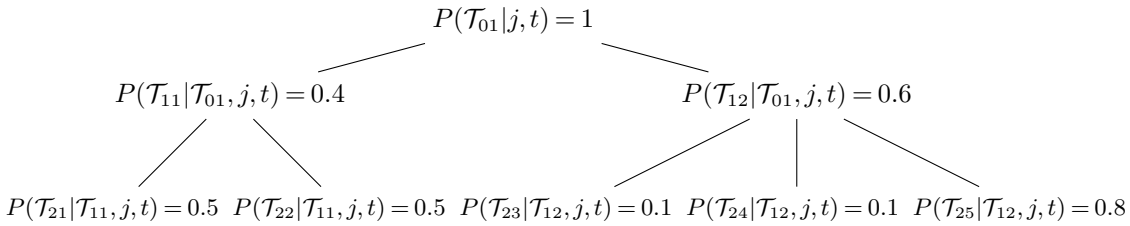

Figure 5 **Probability $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}),j)$ for Each Industry in the Example ICS**

The objective of hierarchical assignment is to compute $P(\mathcal{T}_{li}|j,t)$, the probability that in period t, firm j is assigned to industry $\mathcal{T}_{li}$ among all the industries at level l. In accordance with Bayes' theorem, we have

$$P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}),j,t)=\frac{P(\mathcal{P}(\mathcal{T}_{li})|\mathcal{T}_{li},j,t)P(\mathcal{T}_{li}|j,t)}{P(\mathcal{P}(\mathcal{T}_{li})|j,t)}. \tag{21}$$

According to the hierarchy constraint, if firm j is assigned to industry $\mathcal{T}_{li}$, the firm must belong to $\mathcal{T}_{li}$'s parent industry $P(\mathcal{T}_{li})$ as well. Hence, we have $P(\mathcal{P}(\mathcal{T}_{li})|\mathcal{T}_{li},j,t)=1$. Accordingly, Equation 21 can be rewritten as

$$P(\mathcal{T}_{li}|j,t)=P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}),j,t)P(\mathcal{P}(\mathcal{T}_{li})|j,t). \tag{22}$$

By applying Equation 22 to the last term in Equation 22, we have

$$P(\mathcal{P}(\mathcal{T}_{li})|j,t)=P(\mathcal{P}(\mathcal{T}_{li})|\mathcal{P}^2(\mathcal{T}_{li}),j,t)P(\mathcal{P}^2(\mathcal{T}_{li})|j,t).$$

Thus, $P(\mathcal{T}_{li}|j,t)$ can be expanded recursively to the root level of the ICS:

$$\begin{aligned} P(\mathcal{T}_{li}|j,t) &= P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}),j,t)P(\mathcal{P}(\mathcal{T}_{li})|\mathcal{P}^2(\mathcal{T}_{li}),j,t)P(\mathcal{P}^2(\mathcal{T}_{li})|j,t) \\ &= \dots \\ &= \left(\Pi_{m=0}^{l-1} P(\mathcal{P}^m(\mathcal{T}_{li})|\mathcal{P}^{m+1}(\mathcal{T}_{li}),j,t)\right) P(\mathcal{T}_{01}|j,t). \end{aligned}$$

Recall that $P(\mathcal{T}_{01}|j,t)=1$, because $\mathcal{T}_{01}$ denotes the entire ICS. Therefore, $P(\mathcal{T}_{li}|j,t)$ is factorized as

$$P(\mathcal{T}_{li}|j,t) = \Pi_{m=0}^{l-1} P(\mathcal{P}^m(\mathcal{T}_{li})|\mathcal{P}^{m+1}(\mathcal{T}_{li}),j,t), \tag{23}$$

each factor of which can be computed using Equation 20.

Example 4. Continuation of Example 3. By applying Equation 23, we can compute $P(\mathcal{T}_{25}|j,t)$ as

$$P(\mathcal{T}_{25}|j,t) = P(\mathcal{T}_{25}|\mathcal{T}_{12},j,t)P(\mathcal{T}_{12}|\mathcal{T}_{01},j,t) = 0.8\times 0.6 = 0.48.$$

Hierarchical assignment is distinct from the flat assignment employed by existing industry assignment methods, where a flattened class space is constructed with each class corresponding to an industry at the focal level. Flat assignment only considers industries at the focal level and ignores industries that are above them, thereby neglecting structure-based knowledge. In contrast, hierarchical assignment considers industries across hierarchical levels and incorporates structure-based knowledge into the factorization structure for computing $P(\mathcal{T}_{li}|j,t)$ (i.e., Equation 23).

4.4. DeepIA

DeepIA is trained with past firm–industry assignments $\mathcal{D}=\{(j,y^{(t,j)}) \mid j\in\mathcal{U},\ y^{(t,j)}\in\mathcal{T}_{l^*},\ t\in 1:T\}$. For each $(j,y^{(t,j)})\in\mathcal{D}$, the representation of firm j in period t is derived using Equation 5. Next, the industry representations in periods $1:T$ are derived by integrating definition-based, assignment-based, and structured-based knowledge through the temporal and spatial aggregation mechanisms. Finally, for each $(j,y^{(t,j)})\in\mathcal{D}$, the probability $P(y^{(t,j)}|j)$ that firm j belongs to industry $y^{(t,j)}$ in period t is computed using Equation 23. Let Φ denote the model parameters of DeepIA, we have

$$\Phi = \{\Theta, \Theta', \Delta, \Gamma\}, \tag{24}$$

where the parameters Θ and Θ' are respectively specified in Equations 12 and 18, Δ denotes the parameters associated with transformation layer e_f in Equation 5, and Γ contains randomly initialized representations of leaf-level assignment-based knowledge (see Section 4.2.1)—specifically, $\Gamma=\{v_{Li}^{(A,t)} \mid i\in\mathcal{T}_L, t=1:T\}$.[13]

[13]We compute the model complexity of DeepIA based on Equation 24. Specifically, Θ is of size $4hd^2+d^2+d=(4h+1)d^2+d$, where $4hd^2$ accounts for the number of parameters of the h attention heads each with four $d\times d$ matrices (i.e., $W^{(Q)},W^{(K)},W^{(V)}$, $W^{(O)}$), d^2 is the size of the $d\times d$ matrix W_1, and d is the length of vector b_1. The parameter set Θ' has the same structure as Θ and therefore is of size $(4h+1)d^2+d$. The transformation layer e_f will be later specified in Section 5.2 as a multi-layer perceptron with two layers, and hence its parameter set Δ is of size $(2d\times d+2d)+(d\times 2d+d)=4d^2+3d$. Lastly, Γ is of size $N_L T d$, where N_L is the number of industries at leaf level L and d is the size of embedding vectors. In conclusion, Φ, the parameter set of DeepIA, is of size $(8h+6)d^2+(N_L T+5)d$.

The model parameters for DeepIA are learned by optimizing the following objective function through mini-batch gradient descent:

$$\Phi^* = \arg\min_{\Phi} \sum_{(j,y^{(t,j)})\in\mathcal{D}} -\log P(y^{(t,j)}|j,t). \tag{25}$$

Algorithm 1 The Training Procedure for DeepIA

Input: Firms' business description documents, focal level industry l^*, past firm–industry assignments $\mathcal{D}$, encoded expert knowledge $(\mathcal{T}_{li}^{(D)}, \mathcal{T}_{li}^{(S)}, \mathcal{T}_{li}^{(A,t)})$ for all $\mathcal{T}_{li} \in \mathcal{T}$ and $t \in 1:T$

Output: Learned model parameters Φ^*

1: Pretrain a DEM on firms' business description documents
2: Pretrain a DEM on $\{\mathcal{T}_{li}^{(D)} | \mathcal{T}_{li} \in \mathcal{T}\}$
3: Initialize Φ ▷ Eq. 24
4: **for** epoch $\in 1:N_{\text{epoch}}$ **do** ▷ N_{epoch}: number of epochs
5: Permute $\mathcal{D}$ and partition it into batches $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_{N_{\text{batch}}}$ ▷ N_{batch}: number of batches
6: **for** $k \in 1:N_{\text{batch}}$ **do**
7: **for** $l \in L-1:1$ **do**
8: Compute $\{v_{li}^{(A,t)} \mid i \in \mathcal{T}_l, t \in 1:T\}$ ▷ Eq. 11
9: Compute $\{v_{li}^{(D)} \mid i \in \mathcal{T}_l\}$ ▷ Eq. 15
10: **for** $\mathcal{T}_{li} \in \mathcal{T}$ **do**
11: Compute $\{v_{li}^{(t)} \mid t \in 1:T\}$ ▷ Eq. 17
12: **for** $(j, y^{(t,j)}) \in \mathcal{B}_k$ **do**
13: Compute $x^{(t,j)}$ ▷ Eq. 5
14: **for** $\mathcal{T}_{li}$ in $\mathcal{T}$ **do**
15: Compute $s(\mathcal{T}_{li}, j, t)$ ▷ Eq. 19
16: Compute $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, t)$ ▷ Eq. 20
17: **for** $\mathcal{T}_{l^*i}$ in $\mathcal{T}_{l^*}$ **do**
18: Compute $P(\mathcal{T}_{l^*i}|j,t)$ ▷ Eq. 23
19: Compute loss $\mathcal{L} = \sum_{(j,y^{(t,j)})\in\mathcal{B}_k} -\log P(y^{(t,j)}|j,t)$
20: Compute gradients $\partial\mathcal{L}/\partial\Phi$
21: Update Φ with gradient descent
22: **return** Φ as Φ^*

The training procedure for finding Φ^* is summarized in Algorithm 1. In lines 1 and 2 of the algorithm, two DEMs are respectively trained on the corpus of firms' business description documents

and the corpus of industry definitions. Line 3 randomly initializes all of the model parameters. Starting from line 4, the algorithm is executed for N_{epoch} epochs. At the beginning of each epoch, the assignment cases in $\mathcal{D}$ are randomly permuted and then partitioned into N_{batch} batches. Each batch $\mathcal{B}_k$ consists of a subset of the assignment cases from $\mathcal{D}$, with the size of the batch specified in Section 5.2. The computation steps between line 6 and line 21 illustrate how the model parameters can be iteratively learned in a batch-by-batch style. At the beginning of each batch, all of the DIR vectors (Section 4.2) are prepared between lines 7 and 11. Then the HA procedure (Section 4.3) is performed between lines 12 and 18 for each assignment case in the batch. Next, the loss of the batch to be minimized is computed at line 19, with the gradients derived in the next line. Lastly, based on the gradient information, the model parameters are updated through gradient descent.

Algorithm 2 The Inference Procedure of DeepIA

Input: Business description document $doc^{(T+1,j)}$ of firm j in period $T+1$ and focal-level industry l^*

Output: Predicted industry assignment $\hat{y}^{(T+1,j)}$

1: Compute DEM($doc^{(T+1,j)}$) and then $x^{(T+1,j)}$ ▷ Eq. 5
2: **for** $\mathcal{T}_{li}$ in $\mathcal{T}$ **do**
3: Compute $s(\mathcal{T}_{li}, j, T+1)$ ▷ Eq. 19
4: Compute $P(\mathcal{T}_{li}|\mathcal{P}(\mathcal{T}_{li}), j, T+1)$ ▷ Eq. 20
5: **for** $\mathcal{T}_{l^*i}$ in $\mathcal{T}_{l^*}$ **do**
6: Compute $P(\mathcal{T}_{l^*i}|j, T+1)$ ▷ Eq. 23
7: $\hat{y}^{(T+1,j)} = \arg\max_{\mathcal{T}_{l^*i} \in \mathcal{T}_{l^*}} P(\mathcal{T}_{l^*i}|j, T+1)$
8: **return** $\hat{y}^{(T+1,j)}$

Once trained, DeepIA can be applied to classify an unassigned firm $j \in \mathcal{U}$ in period $T+1$ into a focal-level industry following the steps in Algorithm 2. Specifically, the representation of the unassigned firm is derived at line 1 based on its most up-to-date business description document. Next, the assignment probabilities are computed between lines 2 and 6. Note that the computation of $s(\mathcal{T}_{li}, j, T+1)$ requires $v_{li}^{(A,T+1)}$, which is not available because the true industry assignment of firm j in period $T+1$ is unknown at the time of prediction. Therefore, we use $s(\mathcal{T}_{li}, j, T)$ as a surrogate for $s(\mathcal{T}_{li}, j, T+1)$. In the last line, the industry assigned to the firm, $\hat{y}^{(T+1,j)}$, is the one with the highest assignment probability at the focal industry level.

5. Empirical Evaluation

We benchmark DeepIA against several prevalent methods on the tasks of assigning firms to industries of two widely used ICSs: North American Industry Classification System (NAICS) and Global Industry Classification Standard (GICS). We report evaluation results with NAICS in this section. Similar evaluation results are obtained using GICS and reported in Appendix A for space consideration.

5.1. Data and Evaluation Procedure

Our evaluation was conducted using public data. Specifically, we acquired a firm universe from the Compustat Company Header History (COMPHIST) data provided by Wharton Research Data Services (WRDS). Each row in COMPHIST records the information of a firm in a particular time period. We focused on the following fields in COMPHIST: GVKEY of a firm, which uniquely identifies a firm in the database, HNAICS of a firm, which gives six-digit NAICS code assigned to a firm, as well as the time period within which the information is valid. For example, one record for firm *Costco Wholesale Corporation* is (GVKEY=29028, HNAICS=452910) and this record is valid in year 2012. That is, *Costco Wholesale Corporation* is assigned to NAICS industry *Warehouse Clubs and Supercenters* (coded as 452910) in year 2012. From COMPHIST, we collected a dataset of firm-industry assignments from year 2012 to year 2016. Each record of the dataset shows the assignment of a firm to an industry in a year. Table 3 reports the summary statistics of the dataset. In this table, column n indicates the number of firm-industry assignments in a year. In the dataset, a firm is only assigned to one industry in a year; hence, n also shows the number of unique firms in a year. For instance, there are 5,322 firm-industry assignments or unique firms in year 2012. Column n_{new} means the number of new firms in a year. A firm is considered new in a year if it does not appear in any previous year(s). For example, out of 5,402 firms in year 2013, 692 are new firms.

Table 3 Summary Statistics of the Firm-Industry Assignment Dataset

Year	n	n_{new}
2012	5,322	-
2013	5,402	692
2014	5,360	421
2015	5,335	434
2016	5,054	258

For each firm-year observation in the firm-industry assignment dataset, we constructed the business description document of the firm as Items 1 and 1A of the firm's 10-K report filed in that year. We employ 10-K reports as the source of business description because Items 1 and 1A of a

firm's 10-K report contain accurate, up-to-date, and rich information about its business activities and associated risk factors (Hoberg and Phillips 2016). Specifically, we collected business description documents from the Stage One 10-X Parse Data, which contain 10-K reports preprocessed by Loughran and Mcdonald (2016).[14] On average, a business description document in our evaluation consists of 10,000 words. As an example, Figure 6 shows an excerpt of the business description document of *Costco Wholesale Corporation* in year 2012.

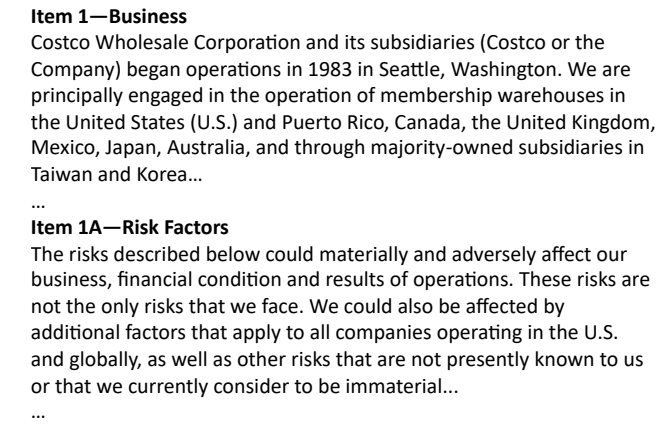
Item 1—Business
Costco Wholesale Corporation and its subsidiaries (Costco or the Company) began operations in 1983 in Seattle, Washington. We are principally engaged in the operation of membership warehouses in the United States (U.S.) and Puerto Rico, Canada, the United Kingdom, Mexico, Japan, Australia, and through majority-owned subsidiaries in Taiwan and Korea…
…
Item 1A—Risk Factors
The risks described below could materially and adversely affect our business, financial condition and results of operations. These risks are not the only risks that we face. We could also be affected by additional factors that apply to all companies operating in the U.S. and globally, as well as other risks that are not presently known to us or that we currently consider to be immaterial…
…

Figure 6 An Excerpt of Costco's Business Description Document in Year 2012

The structure of NAICS as well as the definitions of its industries are revised every five years and the revision occurred in year 2012 (NAICS 2012) covers year 2012 to year 2016. Therefore, we used NAICS 2012 in our evaluation. Table 4 summarizes the structure of NAICS 2012, where l denotes industry level, N_l indicates the number of industries at level l, and D_l means the number of digits for an industry code at level l.[15] For example, there are 1,065 unique industries at level 5 of NAICS 2012 and each industry at this level is denoted by a six-digit code.

Table 4 Structure of NAICS 2012

l	1	2	3	4	5
N_l	20	99	312	713	1,065
D_l	2	3	4	5	6

Having introduced data, we detail our evaluation procedure. Each method (ours or benchmark) takes firm-industry assignment data and business description documents from year 2012 to year T as training data to predict industry assignment for each new firm in year $T+1$ according to NAICS 2012. We focus on predicting industry assignments for new firms because there are no assignment

[14]The data set can be downloaded at https://sraf.nd.edu/data/stage-one-10-x-parse-data/.

[15]The official website for NAICS 2012 can be accessed at https://www.census.gov/naics/?58967?yearbck=2012, which contains its detailed structural information and definitions of its industries.

records for these firms in the training data. To fine-tune hyperparameters of each method, we use assignment data of new firms in year T as validation data. Take $T = 2015$ as an example. Training data for this evaluation contain firm-industry assignments from year 2012 to year 2015 as well as business description document for each firm-year observation in the same time period. Validation data consist of industry assignments of new firms in year 2015. The objective of the evaluation is to predict industry assignment for each new firm in year 2016. The performance of a method is evaluated by comparing its predicted industry assignments against true industry assignments in year $T+1$ based on commonly used metrics: accuracy and macro-F1 (Narasimhan et al. 2016, Wood et al. 2017). Specifically, accuracy is the percentage of correctly classified firms. Let TP_i be the number of firms that are predicted belonging to industry $\mathcal{T}_{l^*i}$ and actually belong to the industry, $i = 1, 2, \ldots, N_{l^*}$, where l^* denotes the focal level and N_{l^*} is the number of industries at the level. Similarly, let FP_i be the number of firms that are predicted belonging to industry $\mathcal{T}_{l^*i}$ but actually do not belong to the industry and FN_i be the number of firms that are predicted not belonging to industry $\mathcal{T}_{l^*i}$ but actually belong to the industry. Precision p_i, recall r_i, and F1-score $F1_i$ for industry $\mathcal{T}_{l^*i}$ are defined as:

$$p_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \quad r_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \quad F1_i = \frac{2p_i r_i}{p_i + r_i}.$$

Macro-F1 is calculated as the mean of F1-scores across industries at the focal level:

$$\text{macro-F1} = \frac{1}{N_{l^*}} \sum_{i \in \mathcal{T}_{l^*}} F1_i.$$

5.2. Benchmark Methods

We benchmark our proposed method, DeepIA, against representative existing industry assignment methods as well as prevalent methods that can be adapted for industry assignment. As discussed in Section 2.1, existing methods employ machine or deep learning models for industry assignment. Therefore, we compare our method against the state-of-the-art industry assignment method proposed by Wood et al. (2017), who design a deep learning model, i.e., multi-layer perceptron, to assign firms to industries. We also consider other representative industry assignment methods. One method is developed based on support vector machine (SVM), a classical machine learning model (Roelands et al. 2010). Another method is developed based on ULMFiT (Howard and Ruder 2018), which can be finetuned to do industry assignment (Tagarev et al. 2019). While existing industry assignment methods neglect structure-based and definition-based knowledge, some recently developed classification models can be adapted to process structure-based or definition-based knowledge for industry assignment, as reviewed in Section 2.2. In this regard, we benchmark our method against a widely applied hierarchical classification method (Ceci and Malerba 2007, Silla and Freitas 2011). To incorporate structure-based knowledge into classification, this method learns a SVM

classifier for each node above the focal level in a tree-shaped industry hierarchy. Industry assignment is then conducted progressively in a top-down manner. First, the root node classifier predicts the most probable level one industry for a firm. The classier at the node corresponding to the predicted industry then selects its most probable child industry for the firm. The latter step is repeated until an industry at the focal level is assigned to the firm. In addition, we benchmark our method against a state-of-the-art label embedding method (Pappas and Henderson 2019), which can be adapted to encode definition-based knowledge. This method represents industry definitions and business description documents of firms as numerical vectors. The compatibility score between a firm and an industry is then estimated using a neural network that takes their representation vectors as inputs. Table 5 summarizes the methods compared in our evaluation.

Table 5 Methods Compared in Our Evaluation

Method	Notes
DeepIA	Our proposed method
SVM-IA	Representative industry assignment method developed based on support vector machine (SVM) (Roelands et al. 2010)
MLP-IA	Representative industry assignment method developed based on multi-layer perceptron (MLP) (Wood et al. 2017)
ULMFiT-IA	Representative industry assignment method developed based on ULMFiT (Tagarev et al. 2019)
HC-IA	Widely applied hierarchical classification method adapted for industry assignment (Silla and Freitas 2011)
LE-IA	State-of-the-art label embedding method adapted for industry assignment (Pappas and Henderson 2019)

Next, we discuss the implementation details of these methods. To classify a firm, each method took the firm's business description document as input and represented the document with a document embedding model (DEM). The DEM used by all the methods, except for RNN-IA, is the Doc2Vec model (Le and Mikolov 2014). Doc2Vec was selected because it is particularly suitable for summarizing semantics of long documents, considering that the average length of a business description document is 10,000 words.[16] ULMFiT-IA used its own DEM as described in Tagarev et al. (2019). Definition-based knowledge used by DeepIA and LE-IA was also embedded by applying Doc2Vec to the corpus of industry definitions. Hyperparameters of each method were tuned and determined using validation data. For DeepIA, we set the embedding size d to 400. The transformation layer e_f in Equation 5 was implemented as a multi-layer perceptron with two layers.

[16] Pappas and Henderson (2019) embed short documents with a sequential compositional neural network, which is not suitable to represent long business description documents in our study.

The sizes of its input layer, hidden layer, and output layer were 400, 800, and 400 respectively; the hidden layer had a dropout rate of 0.5 and a ReLU activation function. DeepIA was trained using the Adam optimizer (Kingma and Ba 2015) with a learning rate of 0.001 and a batch size of 500. MLP-IA had three hidden layers of sizes 640, 4096, and 4096 respectively. For ULMFiT-IA, we follow the steps by Tagarev et al. (2019) and use 400 as the embedding size. For LE-IA, we set the embedding size for both firm and industry representations to 400. Both SVM-IA and HC-IA employed SVM with the stochastic gradient descent linear kernel.

5.3. Evaluation Results

Following the evaluation procedure, we set $T = 2015$ and focal industry level $l^* = 3$. That is, in an experimental run, each method (ours or benchmark) took firm-industry assignments and firms' business description documents from year 2012 to year 2015 as training data and learned a model to classify each new firm in year 2016 into a level three NAICS industry. Table 6 reports the average accuracy and macro-F1 for each method across 20 experimental runs. We also list the percentage improvement by our method over a benchmark method in parentheses. As shown in the table, DeepIA achieves the best performance among all the compared methods in both metrics. In terms of accuracy, 68% of new firms in year 2016 are correctly classified by DeepIA, which outperforms the best performing benchmark method by 7.9%. Moreover, the average macro-F1 of our method is 0.26, which is 8.3% higher than that of the best performing benchmark method. We applied the t-test to the performance data over 20 experimental runs and noted that our method significantly outperformed each benchmark method in both metrics ($p < 0.01$). We note that the values of macro-F1 are lower than those of accuracy in Table 6. The reason is that macro-F1 is a simple average of F1-scores across industries without considering the number of firms in an industry while accuracy takes industry sizes into account. Consequently, misclassifying firms in small-sized industries has significantly negative impact on macro-F1 but relatively small impact on accuracy. We also note that macro-F1 results in our evaluation are at similar level as those reported in the literature, e.g., Wood et al. (2017).

Table 6 Performance Comparison between DeepIA and Benchmark Methods ($T = 2015$ and $l^* = 3$)

	Accuracy	Macro-F1
DeepIA	0.68	0.26
HC-IA	0.63 (7.9%)	0.23 (13.0%)
LE-IA	0.63 (7.9%)	0.24 (8.3%)
ULMFiT-IA	0.62 (9.7%)	0.22 (18.2%)
MLP-IA	0.61 (11.4%)	0.21 (23.8%)
SVM-IA	0.60 (13.3%)	0.22 (18.2%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

Among the compared methods, three existing industry assignment methods, SVM-IA, MLP-IA and ULMFiT-IA perform the worst because they ignore definition-based and structure-based knowledge as well as the time-specificity of assignment-based knowledge. In comparison to these three methods, both HC-IA and LE-IA consider additional type of knowledge (i.e., structure-based or definition-based knowledge), thereby achieving better performance. Our method attains the best performance because it integrates all three types of expert knowledge for industry assignment and takes into account the time-specificity of assignment-based knowledge, which are realized through its two methodological innovations: dynamic industry representation and hierarchical assignment. Therefore, these methodological innovations eventually lead to the superior performance of our method.

To ensure the robustness of our evaluation results, we performed additional evaluations by varying focal industry level l^* from 2 to 5. Table 7 Panel A and Panel B respectively report the average accuracy and macro-F1 for each method over 20 experimental runs. As reported, DeepIA remains the best method across investigated focal industry levels in both metrics. Particularly, DeepIA outperforms the best performing benchmark method by 5.6% to 9.3% in accuracy and by 8.3% to 12.0% in macro-F1.[17] The superiority of our method over each benchmark method is also statistically significant ($p < 0.01$) across investigated focal industry levels.

Table 7 Performance Comparison between DeepIA and Benchmark Methods ($T = 2015$)

	$l^* = 2$	$l^* = 3$	$l^* = 4$	$l^* = 5$
Panel A: Accuracy				
DeepIA	0.70	0.68	0.57	0.46
HC-IA	0.63 (11.1%)	0.63 (7.9%)	0.54 (5.6%)	0.43 (7.0%)
LE-IA	0.64 (9.3%)	0.63 (7.9%)	0.53 (7.5%)	0.43 (7.0%)
ULMFiT-IA	0.63 (11.1%)	0.62 (9.7%)	0.53 (7.5%)	0.42 (9.5%)
MLP-IA	0.62 (12.9%)	0.61 (11.4%)	0.51 (11.8%)	0.42 (9.5%)
SVM-IA	0.63 (11.1%)	0.60 (13.3%)	0.52 (9.6%)	0.43 (7.0%)
Panel B: Macro-F1				
DeepIA	0.33	0.26	0.25	0.28
HC-IA	0.30 (10.0%)	0.23 (13.0%)	0.22 (13.6%)	0.25 (12.0%)
LE-IA	0.28 (17.9%)	0.24 (8.3%)	0.22 (13.6%)	0.25 (12.0%)
ULMFiT-IA	0.27 (22.2%)	0.22 (18.2%)	0.22 (13.6%)	0.25 (12.0%)
MLP-IA	0.26 (26.9%)	0.21 (23.8%)	0.23 (8.7%)	0.25 (12.0%)
SVM-IA	0.28 (17.9%)	0.22 (18.2%)	0.20 (25.0%)	0.24 (16.7%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

[17] As focal industry level l^* varies from 2 to 5, the number of industries increases dramatically. Consequently, as l^* increases, it becomes more difficult to accurately classify a firm to an industry and the accuracy of each method decreases (as reported in Table 7 Panel A). We do not observe exactly the same trend for macro-F1 because it is calculated differently from accuracy. Macro-F1 is a simple average of F1-scores over industries without considering the number of firms in an industry whereas accuracy takes industry sizes into consideration.

We also evaluated the performance of the methods for another year. Specifically, we conducted an evaluation by setting $T = 2013$ and focal industry level $l^* = 3$. The average accuracy and macro-F1 of each method across 20 experimental runs in this evaluation are reported in Table 8. Again, our method performs the best among all the investigated methods. It outperforms the best performing benchmark method by 9.4% in accuracy and 13.6% in macro-F1 and its superiority over each benchmark method is statistically significant ($p < 0.01$).

Table 8 Performance Comparison between DeepIA and Benchmark Methods ($T = 2013$ and $l^* = 3$)

	Accuracy	Macro-F1
DeepIA	0.58	0.25
HC-IA	0.52 (11.5%)	0.22 (13.6%)
LE-IA	0.53 (9.4%)	0.21 (19.0%)
ULMFiT-IA	0.53 (9.4%)	0.22 (13.6%)
MLP-IA	0.51 (13.7%)	0.22 (13.6%)
SVM-IA	0.52 (11.5%)	0.22 (13.6%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

To further investigate the performance of DeepIA in different contexts, we include the following analysis in appendices. In Appendix A, we conducted an additional evaluation with a different ICS, Global Industry Classification Standard (GICS). We obtained largely similar evaluation results and reported them in the appendix. In Appendix B, we evaluated the performance of an industry assignment method under different degrees of automation. In other words, it is allowed to have some percentage of firms manually assigned, which emulates the real world deployment of an industry assignment method. In Appendix C, we investigated the outcome of one additional evaluation metric that accounts for partially correct predictions. In Appendix D, we considered alternative designs of the temporal aggregation mechanism based on methods in the recurrent neural network family. In summary, the analysis in this section and in appendices demonstrate the superiority of our method over benchmark methods in a variety of evaluation settings, which further corroborates the effectiveness of simultaneously considering the three types of expert knowledge for industry assignment.

5.4. Performance Analysis

Our method features two methodological novelties: dynamic industry representation (DIR) and hierarchical assignment (HA). To evaluate the contribution of each novelty to the performance of our method, we firstly drop the HA component from DeepIA. As defined in Equation 23, the HA component computes the probability $P(\mathcal{T}_{li}|j,t)$ of assigning firm j to industry $\mathcal{T}_{li}$ in period t

recursively according to the hierarchical structure of an ICS. By dropping this component, the computation of $P(\mathcal{T}_{li}|j,t)$ ignores the hierarchical structure of the ICS, and focuses only on industries at level l, i.e., industries in $\mathcal{T}_l$. Accordingly, $P(\mathcal{T}_{li}|j,t)$ is computed as

$$P(\mathcal{T}_{li}|j,t) = \frac{s(\mathcal{T}_{li},j,t)}{\sum_{i' \in \mathcal{T}_l} s(\mathcal{T}_{li'},j,t)}.$$

We refer to the resulted method without the HA component as DeepIA-H.

Next, we further drop the DIR component. The objective of this component is to produce a dynamic industry representation $v_{li}^{(t)}$ for each industry based on its definition, structure, and assignment-based knowledge. By dropping this component, $v_{li}^{(t)}$ becomes a static industry representation, v_{li}, which is treated as model parameters. We refer to the resulted method with both the DIR and HA components dropped as DeepIA-HD.

The performance difference between DeepIA and DeepIA-H reveals the contribution of the HA component to the performance of DeepIA and the performance difference between DeepIA-H and DeepIA-HD uncovers the contribution of the DIR component. To investigate the contribution of each component, we conducted experiments with the main experimental setting (i.e., year $T = 2015$ and focal industry level $l^* = 3$). Table 9 compares the accuracy of the three methods. As reported

Table 9 Ablation Analysis of DeepIA ($T = 2015$ and $l^* = 3$)

	Accuracy	Improvement by DeepIA	Improvement by DeepIA-H
DeepIA	0.68		
DeepIA-H	0.66	0.02	
DeepIA-HD	0.61	0.07	0.05

in the table, adding the DIR component to DeepIA-HD improves the accuracy by 0.05 and leads to DeepIA-H, while further adding the HA component to DeepIA-H improves the accuracy by 0.02 and leads to DeepIA. For the accuracy improvement by DeepIA over DeepIA-HD (i.e., 0.07), 28.6% of it is attributed to the HA component and the rest 71.4% is contributed by the DIR component. We also conducted experiments using the metric of Macro-F1, and found that the HA component contributes 11.6% of the Macro-F1 improvement by DeepIA over DeepIA-HD and the DIR component accounts for the remaining 88.4%. In conclusion, both the DIR and HA components contribute to the superior performance of DeepIA while DIR contributes more than HA.

5.5. A Case Study

Having demonstrated the superior performance of our method over benchmarks, it is interesting to show economic value that could be harvested from such superior performance. To that end, we conducted a case study of tax filing by firms. According to a report by Penn Wharton Budget

Model (PWBM), the U.S. statutory corporate tax rate as of December 15, 2017 is 35%; but due to various tax deduction policies, the effective tax rate (ETR) paid by a firm is usually lower.[18] Moreover, ETR varies across industries because many tax deduction policies are industry-specific. We obtained year 2013 ETR for each level one NAICS 2012 industry from PWBM and tabulated them in Table 10.[19] For example, according to the table, year 2013 effective tax rate for a firm in the mining industry is 15.64%.

Table 10 Year 2013 Effective Tax Rate (ETR) for each NAICS 2012 Level One Industry

Agriculture, forestry, fishing, and hunting	Mining	Utilities	Construction
23.82%	15.64%	24.54%	24.61%
Manufacturing	Wholesale trade	Retail trade	Transportation and warehousing
15.39%	23.73%	25.41%	26.28%
Information	Finance and insurance	Real estate, rental, leasing	Professional, scientific, technical services
20.57%	24.20%	24.43%	23.05%
Management of companies	Administrative, waste management services	Educational services	Health care and social assistance
14.38%	23.37%	26.78%	27.52%
Arts, entertainment, and recreation	Accommodation and food services	Other services	
23.95%	14.07%	23.73%	

We leveraged the ETR data to compute the difference between the tax amount a firm should have paid according to its true level one NAICS industry and what it would pay assuming that it filed tax according to its level one NAICS industry predicted by an industry assignment method. Formally, for firm j, let T_j be its true year 2013 ETR, $\hat{T}_j$ denote its year 2013 ETR according to its industry predicted by an industry assignment method, and R_j be its taxable income in year 2013. We then calculated the misclassification cost MC of a method as the average tax difference due to its classification errors:

$$MC = \frac{1}{|\mathcal{U}_I|} \sum_{j \in \mathcal{U}_I} |T_j - \hat{T}_j| R_j$$

where $\mathcal{U}_I$ denotes the set of investigated firms and $|\mathcal{U}_I|$ is the number of firms in the set. In this evaluation, we used a firm's income before extraordinary items to approximate its taxable income.

Using firm-industry assignments and firms' business description documents in year 2012 as training data, we trained each method listed in Table 5 to classify new firms in year 2013 to level one

[18] https://budgetmodel.wharton.upenn.edu/issues/2017/12/15/effective-tax-rates-by-industry

[19] NAICS 2012 contains twenty level one industries. However, PWBM only provides ETR data for nineteen of them, as reported in Table 10.

NAICS industries. Among 692 new firms in year 2013, 310 firms can be linked with valid income and ETR data and were used to compute MC. Firms with negative income or not belonging to any industry in Table 10 (i.e., no ETR data) were dropped. The average taxable income of investigated firms is \$573 million. Table 11 reports the misclassification cost MC of each method in year 2013.[20] Among the compared methods, DeepIA incurs the lowest misclassification cost. Compared to the benchmark methods, DeepIA reduces misclassification cost by a range of 11.70% to 20.09%. In monetary amount, it reduces tax difference by a range of \$0.51 million per firm to \$0.96 million per firm.

Table 11 Misclassification Cost of Each Method

	MC(in millions)	Cost Reduction by DeepIA
DeepIA	\$3.82	
HC-IA	\$4.71	18.78%
LE-IA	\$4.33	11.70%
ULMFiT-IA	\$4.76	19.75%
MLP-IA	\$4.78	20.09%
SVM-IA	\$4.71	18.78%

6. Discussion and Conclusions

Industry assignment is fundamental to a large number of critical business practices, ranging from operations and strategic decision making by firms to economic analyses by government agencies. Existing industry assignment methods utilize only assignment-based knowledge to classify firms into industries and overlook definition-based and structure-based knowledge, although all three types of knowledge are essential for effective industry assignment. To address this limitation of existing methods, we propose a novel method that seamlessly integrates all three types of knowledge for industry assignment. Furthermore, our method considers the time specificity of assignment-based knowledge, which is also neglected by existing methods. Through extensive empirical evaluations with two widely used ICSs, we demonstrate the superiority of our method over representative existing industry assignment methods as well as prevalent classification methods that can be adapted for industry assignment. Our study contributes to the extant literature in two ways. First, our work belongs to the computational genre of design science research, which is concerned with developing computational methods to solve business and societal problems and aims to make methodological contributions (Rai 2017, Padmanabhan et al. 2022). In this regard, the key innovations of our proposed method—dynamic industry representation and hierarchical assignment—constitute the methodological contributions of our study. Second, our study contributes to Fintech research with a novel method that effectively solves a foundational financial problem.

[20] Because the focal industry level is one, HC-IA reduces to SVM-IA and performs the same as SVM-IA.

6.1. Research Implications

This study has implications for predictive and prescriptive analytics research. Over the years, methods that predict future actions (i.e., predictive analytics) or specify optimal decisions (i.e., prescriptive analytics) have been developed to solve problems in a diverse set of domains such as health care, social networks, and Fintech (Abbasi et al. 2012, Li et al. 2016, Fang and Hu 2018, Fang et al. 2021). Our study is particular useful for those models that rely solely on prior instance-class assignments but ignore other relevant information such as the time specificities of these assignments, the definitions of the classes, and the structural relationships among the classes. For instance, a problem that has a similar structure to the industry assignment problem is patent classification, which assigns a patent to a patent class (Gomez and Moens 2014). Three types of expert knowledge are informative for patent classification: historical patent classifications with time-stamps (i.e., dynamic assignment-based knowledge), hierarchical structure of patent classes (i.e., structure-based knowledge), and textual descriptions of patent classes (i.e., definition-based knowledge). Although structure-based knowledge and static assignment-based knowledge have been considered by prior patent classification methods (Gomez and Moens 2014, Aroyehun et al. 2021), definition-based knowledge and time specificities of assignment-based knowledge are overlooked by these methods. In this regard, our study informs future research about how to operationalize these three types of knowledge and incorporate them into a patent classification model. Specifically, we can encode the three types of knowledge for patent classification with Equations 2, 3, 4 respectively and embed a patent filing with Equation 5. Next, by applying the design of dynamic industry representation (DIR) and hierarchical assignment (HA) to patent classification, we can formulate dynamic patent class representation by embedding each patent class as a sequence of time-specific vectors that integrate the three types of knowledge, and assign a patent to a path of patent classes in accordance with the hierarchical structure entailed by the tree-shaped arrangement of patent classes.

As another example, consider the problem of link recommendation for online social networks of which the goal is to recommend friendship links to currently unlinked users (Li et al. 2016). From the perspective of a focal user, we can construct a sequence of historical friendship links established by the user. This sequence bears an analogy to the dynamic assignment-based knowledge by treating the focal user as a class and the friends made by the user in the past as instances assigned to the class. One consequence of this perspective is that each user now has two roles: a role of class when she acts as a focal user, and a role of instance when she acts as a friend of other focal users. Another consequence is that an instance can be assigned to multiple classes because multiple focal users can make the same friend. In addition to the dynamic assignment-based knowledge, definition-based knowledge about a focal user can also be built based on the user's profile. However, the definition of

classes (focal users) and the description of instances (friends) now come from the same source, and can be more general than text. Lastly, a hierarchical community detection algorithm (Li et al. 2020) can be employed to cluster focal users into hierarchical groups, which establishes the structure-based knowledge for link recommendation. In this way, while a focal user serves as a class, the communities the focal user belongs to can be regarded as meta classes. Although the assignment-based, definition-based, and structure-based knowledge are not generated by "experts", our study still suggests a novel model architecture for the link recommendation problem. Specifically, applying the design of DIR, we can construct dynamic focal user representation by embedding each focal user as a sequence of time-specific vectors that integrates the three types of knowledge about the user. Next, given that an instance can be assigned to multiple classes, the HA procedure needs to be adjusted accordingly to solve a multi-label classification problem instead of a multi-class classification problem. This can be done by normalizing each compatibility score between an instance and a class to a probability using the sigmoid function (Goodfellow et al. 2016).

6.2. Practical Implications

Our study also has several implications for business. The analysis in Section 5.5 demonstrates the financial impact of industry assignment on firms and the government. Indeed, industry assignment can significantly affect firms' operating costs in a variety of business settings, including raising capital and purchasing commercial insurance.[21] For example, an incorrect classification into a higher risk industry than the one in which a firm is actually operating can bring down the firm's business credit score, which in turn leads to more stringent loan covenants and higher interest rates. Therefore, developing methods that can produce accurate industry assignments is beneficial for both firms and the government. Specifically, we suggest that an automated industry assignment method like DeepIA can be deployed with a proper production rate (discussed in Appendix B) and serve as an assistive system in the following way. Domain experts can trust its assignment predictions with top assignment scores without further intervention while focusing on verifying and correcting those predictions with low assignment scores.[22] By intertwining the automatic and manual industry assignment procedures, misclassification cost can be managed with minimum human effort.

In addition, our method characterizes the evolution of an industry by representing it as a sequence of time-specific vectors. Comparisons of vectors within an industry sequence or across industry sequences could shed light on how to revise the structure of an ICS in response to the changing

[21] Please see https://www.insureon.com/small-business-insurance/cost and https://www.nav.com/blog/is-your-naics-code-costing-you-money-19128/.

[22] A firm's industry assignment score is the probability of its most-likely assigned industry predicted by DeepIA.

business landscape. For instance, if the vectors for two industries gradually become similar over time, it might be appropriate to merge them into one industry. Finally, the three types of expert knowledge emphasized by our method are not unique to industry assignment. Patent classification is another important application where these types of expert knowledge exist and there is an urgent need to develop automated methods to classify the sheer number of patent applications into hierarchically organized patent classes (Gomez and Moens 2014). Therefore, our method is general and can be adapted to other critical applications where the consideration of these types of expert knowledge is essential to model performance.

6.3. Limitations and Future Research

Our study has limitations and can be extended in multiple ways. First, our method represents a firm using only textual data in its business description document. Numerical data about a firm, such as financial ratios, could also convey useful information for its industry assignment because these data are correlated with the industry characteristics of the firm (Gupta and Huefner 1972). Future research could explore whether the incorporation of relevant numerical data into our method can further enhance its performance. Second, the DEM employed by our method is pretrained using the corpus of firms' business description documents rather than learned together with other components of DeepIA. It is worth investigating whether there exists an alternative source of firms' business descriptions that convey rich information in short length, such that the DEM can be integrated into DeepIA in an end-to-end training manner. Third, our evaluation focuses on public firms because their business description documents and industry assignment data are publicly available. It would be interesting to evaluate our method using business description documents and industry assignment data for private firms, which are usually purchased from third-party data vendors (Wood et al. 2017). Considering the sheer volume of private businesses established every year, industry assignment for private firms could achieve high classification accuracy by leveraging more training data. Fourth, our method learns industry representations using observed assignment-based knowledge but ignores future firm–industry assignment patterns. Future work could predict these patterns and represent industries with both observed assignment-based knowledge and predicted firm–industry assignment patterns. In doing so, our method could be more effective in classifying firms in a future time period. Lastly, the learning objective of DeepIA (Equation 25) is sequential in nature. As a result, if the method makes an incorrect decision at an upper industry level, it will misclassify a firm at lower levels. Therefore, it might be beneficial to conduct multiple label classification, which encourages the method to correctly classify a firm at all industry levels from the root to the focal one. This can be done by leveraging the factorization structure of Equation 23, where level-specific decision weights can be imposed to emphasize the importance of upper level decisions.

References

Abbasi A, Albrecht C, Vance A, Hansen J (2012) MetaFraud: A Meta-Learning Framework for Detecting Financial Fraud. *MIS Quarterly* 36(4):1293–1327.

Ahern KR, Harford J (2014) The Importance of Industry Links in Merger Waves. *The Journal of Finance* 69(2):527–576.

Aroyehun ST, Angel J, Majumder N, Gelbukh A, Hussain A (2021) Leveraging label hierarchy using transfer and multi-task learning: A case study on patent classification. *Neurocomputing* 464:421–431.

Bao Y, Datta A (2014) Simultaneously Discovering and Quantifying Risk Types from Textual Risk Disclosures. *Management Science* 60(6):1371–1391.

Bhojraj S, Lee CMC, Oler DK (2003) What's my line? A comparison of industry classification schemes for capital market research. *Journal of Accounting Research* 41(5):745–774.

Bizjak J, Lemmon M, Nguyen T (2011) Are all CEOs above average? An empirical analysis of compensation peer groups and pay design. *Journal of Financial Economics* 100(3):538–555.

Bonaime A, Gulen H, Ion M (2018) Does policy uncertainty affect mergers and acquisitions? *Journal of Financial Economics* 129(3):531–558.

Cao J, Liang H, Zhan X (2019) Peer effects of corporate social responsibility. *Management Science* 65(12):5487–5503.

Cavaglia S, Brightman C, Aked M (2000) The Increasing Importance of Industry Factors. *Financial Analysts Journal* 56(5):41–54.

Ceci M, Malerba D (2007) Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems* 28(1):37–78.

Chen BC, Creecy RH, Appel MV (1993) On Error Control of Automated Industry and Occupation Coding. *Journal of Official Statistics* 9(5):729–745.

Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. *Twenty-first international conference on Machine learning - ICML '04*, 27.

Fang X, Gao Y, Hu PJH (2021) A Prescriptive Analytics Method for Cost Reduction in Clinical Decision Making. *MIS Quarterly* 45(1):83–115.

Fang X, Hu PJH (2018) Top persuader prediction for social networks. *MIS Quarterly* 42(1):63–82.

Gao H, He J, Chen K (2020) Exploring Machine Learning Techniques for Text-Based Industry Classification. SSRN Scholarly Paper ID 3640205, Social Science Research Network.

Goldstein I, Jiang W, Karolyi GA (2019) To FinTech and Beyond. *The Review of Financial Studies* 32(5):1647–1661.

Gomez JC, Moens MF (2014) A Survey of Automated Hierarchical Classification of Patents. Paltoglou G, Loizides F, Hansen P, eds., *Professional Search in the Modern World: COST Action IC1002 on Multilingual and Multifaceted Interactive Information Access*, 215–249.

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

Goodman DA, Peavy JW (1983) Industry Relative Price-Earnings Ratios as Indicators of Investment Returns. *Financial Analysts Journal* 39(4):60–66.

Gupta MC, Huefner RJ (1972) A Cluster Analysis Study of Financial Ratios and Industry Characteristics. *Journal of Accounting Research* 10(1):77–95.

Gweon H, Schonlau M, Kaczmirek L, Blohm M, Steiner S (2017) Three Methods for Occupation Coding Based on Statistical Learning. *Journal of Official Statistics* 33(1):101–122.

He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hendershott T, Zhang MX, Zhao JL, Zheng E (2017) Call for Papers—Special Issue of Information Systems Research Fintech – Innovating the Financial Industry Through Emerging Information Technologies. *Information Systems Research* 28(4):885–886.

Hoberg G, Phillips G (2016) Text-based network industries and endogenous product differentiation. *Journal of Political Economy* 124(5):1423–1465.

Howard J, Ruder S (2018) Universal Language Model Fine-tuning for Text Classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339.

Hu D, Zhao L, Hua Z, Wong M (2012) Network-Based Modeling and Analysis of Systemic Risk in Banking Systems. *MIS Quarterly* 36(4):1269.

Jaffe AB (1986) Technological opportunity and spillovers of R&D: Evidence from firms' patents, profits, and market value. *The American Economic Review* 76(5):984–1001.

Jung Y, Yoo J, Myaeng SH, Han DC (2008) A Web-Based Automated System for Industry and Occupation Coding. *Web Information Systems Engineering - WISE 2008*, 443–457.

Kahle KM, Walkling RA (1996) The impact of industry classifications on financial research. *Journal of Financial and Quantitative Analysis* 31(3):309.

Kearney AT, Kornbau ME (2005) An Automated Industry Coding Application for New U.S. Business Establishments. *Proceedings of the American Statistical Association.*

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. *International Conference on Learning Representations.*

Koller D, Sahami M (1997) Hierarchically Classifying Documents Using Very Few Words. *Proceedings of the Fourteenth International Conference on Machine Learning*, 170–178.

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 1188–1196.

Lee CMC, Ma P, Wang CCY (2015) Search-based peer firms: Aggregating investor perceptions through internet co-searches. *Journal of Financial Economics* 116(2):410–431.

Li T, Lei L, Bhattacharyya S, Van den Berge K, Sarkar P, Bickel PJ, Levina E (2020) Hierarchical Community Detection by Recursive Partitioning. *Journal of the American Statistical Association* 0(0):1–18, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2020.1833888.

Li Z, Fang X, Bai X, Sheng ORL (2016) Utility-Based Link Recommendation for Online Social Networks. *Management Science* 63(6):1938–1952.

Lim HS, Lee WKH, Kim HC, Jeong SY, Yu HC (2005) An Automatic Code Classification System by Using Memory-Based Learning and Information Retrieval Technique. *Information Retrieval Technology*, 577–582.

Loughran T, Mcdonald B (2016) Textual analysis in accounting and finance: A survey. *Journal of Accounting Research* 54(4):1187–1230.

McCallum A, Rosenfeld R, Mitchell T, Ng AY (1998) Improving Text Classification by Shrinkage in a Hierarchy of Classes. *Proceedings of the Fifteenth International Conference on Machine Learning*, 359–367.

McGahan AM, Porter ME (1997) How Much Does Industry Matter, Really? *Strategic Management Journal* 18(S1):15–30.

Nam J, Mencía EL, Fürnkranz J (2016) All-in Text: Learning Document, Label, and Word Representations Jointly. *Thirtieth AAAI Conference on Artificial Intelligence*.

Narasimhan H, Pan W, Kar P, Protopapas P, Ramaswamy HG (2016) Optimizing the Multiclass F-Measure via Biconcave Programming. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1101–1106.

Padmanabhan B, Fang X, Sahoo N, Burton-Jones A (2022) Machine Learning in Information Systems Research. *Management Information Systems Quarterly* 46(1):iii–xix.

Pappas N, Henderson J (2019) GILE: A Generalized Input-Label Embedding for Text Classification. *Transactions of the Association for Computational Linguistics* 7:139–155.

Phillips RL, Ormsby R (2016) Industry classification schemes: An analysis and review. *Journal of Business and Finance Librarianship* 21(1):1–25.

Pierre JM (2001) On the Automated Classification of Web Sites. *Linkoping Electronic Articles in Computer and Information Science* 6.

Rai A (2017) Editor's comments: Diversity of design science research. *MIS Quarterly* 41(1):iii–xviii.

Rodrigues F, Pereira FC, Alves A, Jiang S, Ferreira J (2012) Automatic classification of points-of-interest for land-use analysis. *Proceedings of the 4th International Conference on Advanced Geographic Information Systems, Applications, and Services,*, 41–49.

Roelands M, Delden Av, Windmeijer D (2010) Classifying businesses by economic activity using web- based text mining. Discussion Paper, Statistics Netherlands (CBS).

Shi Z, Lee GM, Whinston AB (2016) Toward a better measure of business proximity: Topic modeling for industry intelligence. *MIS Quarterly* 40(4):1035–A53.

Silla CN, Freitas AA (2011) A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22(1):31–72.

Tagarev A, Tulechki N, Boytcheva S (2019) Comparison of Machine Learning Approaches for Industry Classification Based on Textual Descriptions of Companies. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 1169–1175.

Thompson M, Kornbau ME, Vesely J (2012) Creating an automated industry and occupation coding process for the American Community Survey. Technical report.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Wei R, Zheng Q, Dong B, Yang K, He H, Ruan J (2019) ABR-HIC: Attention Based Bidirectional RNN for Hierarchical Industry Classification. *2019 IEEE International Conference on Big Data (Big Data)*, 1527–1536.

Weinberger K, Chapelle O (2009) Large Margin Taxonomy Embedding for Document Categorization. *Advances in Neural Information Processing Systems 21*, 1737–1744.

Weiner C (2005) The impact of industry classification schemes on financial research. *SSRN* .

Wood S, Muthyala R, Jin Y, Qin Y, Rukadikar N, Rai A, Gao H (2017) Automated industry classification with deep learning. *2017 IEEE International Conference on Big Data (Big Data)*, 122–129.

Yazdani M, Henderson J (2015) A Model of Zero-Shot Learning of Spoken Language Understanding. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 244–249.

A. Additional Evaluation with GICS

We conducted an additional robustness check with another popular ICS – Global Industry Classification Standard (GICS), which is jointly developed by Standard and Poor's and Morgan Stanley Capital International (Bhojraj et al. 2003, Phillips and Ormsby 2016). Unlike NAICS, GICS is revised annually. However, we found that the structure of GICS was fairly stable between year 2012 and year 2016. Therefore, we used the version updated in year 2016 in our evaluation. Table A.1 summarizes the structure of GICS 2016. As shown, the industry hierarchy of GICS 2016 contains four levels. There are 171 level four industries, each of which is denoted by a eight-digit code.

Table A.1 Structure of GICS 2016

l	1	2	3	4
N_l	11	25	70	171
D_l	2	4	6	8

We collected a dataset of firm-industry assignments with GICS over the period of year 2012 to year 2016 from COMPHIST. Each record of the dataset shows the assignment of a firm to a level four GICS industry in a year. For example, *Costco Wholesale Corporation* is assigned to level four GICS industry *Hypermarkets and Super Centers* (coded as 30101040) in year 2012. Table A.2 reports the summary statistics of the dataset. In this table, columns n and n_{new} indicate the number of firm-industry assignments and the number of new firms in a year, respectively. In COMPHIST, some firms have NAICS industries but not GICS industries while some other firms carry GICS industries but not NAICS industries. Consequently, the numbers in Table A.2 are different from those in Table 3. For each firm-year observation in the firm-industry assignment dataset, we constructed the business description document of the firm as Items 1 and 1A of the firm's 10-K report filed in that year.

Table A.2 Summary Statistics of the Firm-Industry Assignment Dataset (GICS)

Year	n	n_{new}
2012	6,217	-
2013	5,937	461
2014	5,906	566
2015	5,808	494
2016	5,455	310

Following the evaluation procedure described in Section 5.1, we set $T = 2015$ and focal industry level $l^* = 4$. Accordingly, in an experimental run, each method listed in Table 5 employed firm-industry assignments with GICS from year 2012 to year 2015 as well as firms' business description

documents in the same time period as training data and learned a model to classify each new firm in year 2016 into a level four GICS industry. Table A.3 reports the average accuracy and macro-F1 of each method across 20 experimental runs. As reported, our method achieves the best performance among all the compared methods in both metrics. It surpasses the best performing benchmark method by 7.4% in accuracy and 6.1% in macro-F1. Applying the t-test to the performance data over 20 experimental runs, we noted that our method significantly outperformed each benchmark method in both metrics ($p < 0.01$).

Table A.3 Performance Comparison between DeepIA and Benchmark Methods (GICS, $T = 2015$, $l^* = 4$)

	Accuracy	Macro-F1
DeepIA	0.58	0.35
HC-IA	0.53 (9.4%)	0.32 (9.3%)
LE-IA	0.54 (7.4%)	0.33 (6.1%)
ULMFiT-IA	0.53 (9.4%)	0.32 (9.3%)
MLP-IA	0.52 (11.5%)	0.31 (12.9%)
SVM-IA	0.53 (9.4%)	0.32 (9.3%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

B. Performance of DeepIA under Different Production Rates

An automatic industry assignment (IA) method such as DeepIA can be deployed for real world applications using the concept of production rate, which refers to the percentage of firms that can be classified by the method automatically without the intervention of human experts (Gweon et al. 2017). Specifically, firms are sorted by their industry assignment scores predicted by an automatic IA method, in descending order. One example of a firm's industry assignment score is the probability of its most-likely assigned industry predicted by DeepIA. Given a production rate $PR\%$ of an automatic IA method, the top $PR\%$ firms are automatically assigned by the method while the rest $(1-PR)\%$ firms are manually assigned. In general, there is a trade-off between production rate and the quality of automated assignments. The higher the production rate is, the more errors tend to be in the automatically assigned group due to the inclusion of more lower quality predictions.

Table A.4 tabulates the accuracy of our DeepIA method across different production rates ranging from 50% to 70%, using the main experimental setting (i.e., year $T=2015$ and focal industry level $l^*=3$). As reported in the table, the accuracy of DeepIA increases from 0.73 to 0.81 as we reduce its production rate from 70% to 50%. Following the practice of Kearney and Kornbau (2005), to deploy DeepIA for a real world application, domain experts need to decide an acceptable accuracy level (e.g., 0.78), which in turn determines the production rate of DeepIA (e.g., 60%). We conducted more experiments to compare the accuracy of DeepIA with that of benchmark methods across different production rates. As reported in Table A.5, DeepIA outperforms each benchmark method significantly ($p<0.05$) across different production rates.

Table A.4 Accuracy of DeepIA under Different Production Rates ($T=2015$ and $l^*=3$)

Production Rate	Accuracy
50%	0.81
60%	0.78
70%	0.73

Table A.5 Accuracy of Each Compared Method under Different Production Rates (PR)

	$PR=50\%$	$PR=60\%$	$PR=70\%$
DeepIA	0.81	0.78	0.73
HC-IA	0.78 (3.85%)	0.73 (6.85%)	0.69 (5.80%)
LE-IA	0.77 (5.19%)	0.73 (6.85%)	0.68 (7.35%)
ULMFiT-IA	0.76 (6.58%)	0.71 (9.86%)	0.67 (8.96%)
MLP-IA	0.76 (6.58%)	0.72 (8.33%)	0.67 (8.96%)
SVM-IA	0.77 (5.19%)	0.72 (8.33%)	0.67 (8.96%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

C. Tree-based Misclassification Error

By assigning a firm to an industry in a target ICS, the firm is actually assigned to the corresponding industry path leading from the root node to the assigned industry by tracing the tree-shaped hierarchy of the target ICS. Our evaluation in Section 5.3 treats an assigned firm-industry pair as completely wrong as long as the predicted industry path does not fully overlap with the truly assigned one. However, there exists the possibility that a prediction might be partially correct by being partially aligned with the true industry path. To account for this possibility, we employ an additional metric of the tree-based distance (Dekel et al. 2004, Kosmopoulos et al. 2015). Specifically, if the predicted industry of a firm is $\mathcal{T}_{l^*\hat{y}}$ while its true industry is $\mathcal{T}_{l^*y}$, then the tree-based distance between them is measured as the minimum number of edges from $\mathcal{T}_{l^*\hat{y}}$ to $\mathcal{T}_{l^*y}$ by viewing the given Industry Classification System (ICS) $\mathcal{T}$ as a tree (Kosmopoulos et al. 2015). Consider the example ICS in Figure 2 of the paper. The tree-based distance between $\mathcal{T}_{23}$ and $\mathcal{T}_{24}$ is 2 by following the path composed of edges $\mathcal{T}_{23} \to \mathcal{T}_{12}$ and $\mathcal{T}_{12} \to \mathcal{T}_{24}$, while it is 4 between $\mathcal{T}_{22}$ and $\mathcal{T}_{24}$ by following the path composed of edges $\mathcal{T}_{22} \to \mathcal{T}_{11}$, $\mathcal{T}_{11} \to \mathcal{T}_{01}$, $\mathcal{T}_{01} \to \mathcal{T}_{12}$ and $\mathcal{T}_{12} \to \mathcal{T}_{24}$. Note that the tree-based distance is always zero from a node to itself. In this sense, a partially matched prediction has an error (measured by the tree-based distance) that is decreasing with the degree of overlap between the predicted industry path and the true industry path. Therefore, a more effective industry assignment method has a lower error.

We compared DeepIA with benchmarks in terms of the tree-based distance using the main experimental setting (i.e., year $T = 2015$ and focal industry level $l^* = 3$). For each method in an experimental run, we recorded the average of the tree-based distances across all test assignment cases. The evaluation results averaged across 20 experimental runs are shown in Table A.6. As reported in the table, in comparison to each benchmark method, DeepIA reduces the error measured by the tree-based distance substantially and significantly ($p < 0.01$).

Table A.6 **Tree-based Distance of Each Compared Method ($T = 2015$ and $l^* = 3$)**

	Tree-based Distance	Error Reduction by DeepIA
DeepIA	1.58	
HC-IA	1.77	10.73%
LE-IA	1.81	12.71%
ULMFiT-IA	1.87	15.51%
MLP-IA	1.93	18.13%
SVM-IA	1.97	19.80%

D. Alternative Temporal Aggregation Mechanism

In Section 4.2, we instantiate the temporal aggregation mechanism based on the multi-head self-attention mechanism (Equation 16). Models in the RNN family, such as LSTM, represent another paradigm of temporal aggregation: the information at each timestamp is recursively aggregated into a hidden state vector, which then becomes the context vector of the next timestamp. The recursive nature of the RNN architectures has limitations, such as precluding parallelization among training samples (Vaswani et al. 2017). To address these limitations, the multi-head self-attention mechanism is proposed, and demonstrates superior empirical performance in tasks involving temporal aggregation (Vaswani et al. 2017, Devlin et al. 2019). Consequently, we believe that the multi-head self-attention mechanism is the state-of-the-art architecture for temporal aggregation, and we built our DeepIA model based on it.

To support our choice, we construct a variant of DeepIA by employing LSTM to formulate the temporal aggregation mechanism in lieu of what we report in Section 4.2.2. Specifically, $v_{li}^{(t)}$, originally defined by Equation 16, is now a hidden state vector of a LSTM layer. The LSTM layer takes the sequence $[v_{li}^{(A,1)}, v_{li}^{(A,2)}, \ldots, v_{li}^{(A,T)}]$ (computed in Section 4.2.1) as input, and computes $v_{li}^{(t)}$ as

$$v_{li}^{(t)} = \text{LSTM}(v_{li}^{(t-1)}, v_{li}^{(A,t)})$$

where LSTM is a standard LSTM layer provided by PyTorch. Please refer to `https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html` for the specification of the LSTM layer. To incorporate the definition-based knowledge, we use $v_{li}^{(D)}$ to set the initial hidden state of the LSTM layer. The resulted model is named as DeepIA-LSTM.

Next, we consider a state-of-the-art model in the RNN family: SRU^{++} (Lei 2021), which is developed based on the Simple Recurrent Unit (SRU), a strong substitute for LSTM (Lei et al. 2018). Compared to LSTM, the major design difference of SRU is a light recurrence structure and a highway network imposed on the recursively computed hidden state vectors. Compared to SRU, SRU^{++} transforms the input sequence and then feeds the resulted sequence as the input to a SRU layer. We use SRU^{++} to formulate the temporal aggregation in a similar way to LSTM. Specifically, we have

$$v_{li}^{(t)} = \text{SRU}^{++}(v_{li}^{(t-1)}, v_{li}^{(A,t)})$$

where $v_{li}^{(t)}$ is now a hidden state vector of the SRU^{++} layer. To incorporate the definition-based knowledge, we use $v_{li}^{(D)}$ to set the initial hidden state of the SRU^{++} layer. The resulted model is named as DeepIA-SRU^{++}.

We benchmark DeepIA against DeepIA-LSTM and DeepIA-SRU^{++} using the main experimental setting as reported in Section 5.3 (i.e., year $T = 2015$ and focal industry level $l^* = 3$). As reported

Table A.7 Performance Comparison between DeepIA, DeepIA-LSTM and DeepIA-SRU^{++} ($T = 2015$ and $l^* = 3$)

	Accuracy	Macro-F1
DeepIA	0.68	0.26
DeepIA-SRU^{++}	0.66 (3.0%)	0.25 (4.0%)
DeepIA-LSTM	0.64 (6.3%)	0.24 (8.3%)

Note: The percentage improvement by our method over a benchmark is listed in parentheses.

in Table A.7, DeepIA outperforms DeepIA-SRU^{++} by 3.0% in accuracy and 4.0% in macro-F1. And it surpasses DeepIA-LSTM by 6.3% in accuracy and 8.3% in macro-F1. All the improvements by Deep-IA are statistically significant ($p < 0.01$). We note that the two variants of DeepIA inherit all the elements of DeepIA except for the implementation of the temporal aggregation component. Therefore, the performance advantages of DeepIA over DeepIA-SRU^{++} and DeepIA-LSTM reflect the contribution of implementing the temporal aggregation component based on the multi-head self-attention mechanism.

References

Bhojraj S, Lee CMC, Oler DK (2003) What's my line? A comparison of industry classification schemes for capital market research. *Journal of Accounting Research* 41(5):745–774.

Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. *Twenty-first international conference on Machine learning - ICML '04*, 27.

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Gweon H, Schonlau M, Kaczmirek L, Blohm M, Steiner S (2017) Three Methods for Occupation Coding Based on Statistical Learning. *Journal of Official Statistics* 33(1):101–122.

Kosmopoulos A, Partalas I, Gaussier E, Paliouras G, Androutsopoulos I (2015) Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery* 29(3):820–865.

Lei T (2021) When Attention Meets Fast Recurrence: Training Language Models with Reduced Compute. *arXiv preprint arXiv:2102.12459* .

Lei T, Zhang Y, Wang SI, Dai H, Artzi Y (2018) Simple Recurrent Units for Highly Parallelizable Recurrence. *Empirical Methods in Natural Language Processing (EMNLP)*.

Phillips RL, Ormsby R (2016) Industry classification schemes: An analysis and review. *Journal of Business and Finance Librarianship* 21(1):1–25.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.