Enhancing Multi-field B2B Cloud Solution Matching via Contrastive Pre-training

Haonan Chen
Renmin University of China
Beijing, China
hnchen@ruc.edu.cn

Zhicheng Dou
Renmin University of China
Beijing, China
dou@ruc.edu.cn

Xuetong Hao
Huawei Cloud Computing
Hangzhou, China
haoxuetong@huawei.com

Yunhao Tao
Renmin University of China
Beijing, China
yunhao@ruc.edu.cn

Shiren Song
Renmin University of China
Beijing, China
shiren.song@ruc.edu.cn

Zhenli Sheng
Huawei Cloud Computing
Hangzhou, China
shengzhenli@huawei.com

Abstract

Cloud solutions have gained significant popularity in the technology industry as they offer a combination of services and tools to tackle specific problems. However, despite their widespread use, the task of identifying appropriate company customers for a specific target solution to the sales team of a solution provider remains a complex business problem that existing matching systems have yet to adequately address. In this work, we study the B2B solution matching problem and identify two main challenges of this scenario: (1) the modeling of complex multi-field features and (2) the limited, incomplete, and sparse transaction data. To tackle these challenges, we propose a framework CAMA, which is built with a hierarchical multi-field matching structure as its backbone and supplemented by three data augmentation strategies and a contrastive pre-training objective to compensate for the imperfections in the available data. Through extensive experiments on a real-world dataset, we demonstrate that CAMA outperforms several strong baseline matching models significantly. Furthermore, we have deployed our matching framework on a system of Huawei Cloud. **Our observations indicate an improvement of about 30% compared to the previous online model in terms of Conversion Rate (CVR), which demonstrates its great business value.**

CCS Concepts

• **Information systems** → **Retrieval models and ranking**.

Keywords

Multi-field Matching, Contrastive Learning, Cloud Solutions

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/3637528.3671513

ACM Reference Format:
Haonan Chen, Zhicheng Dou, Xuetong Hao, Yunhao Tao, Shiren Song, and Zhenli Sheng. 2024. Enhancing Multi-field B2B Cloud Solution Matching via Contrastive Pre-training. In *Proceedings of Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, New York, NY, USA, 11 pages. https://doi.org/3637528.3671513

1 Introduction

Cloud solutions, referring to a combination of various cloud-based technologies, tools, and services, have become increasingly popular among companies these years. They are designed to address specific business needs or solve particular problems, such as data storage, application development, customer relationship management (CRM), etc. For solution providers, it is crucial to have an effective matching system that can guide sales teams in identifying potential enterprises that can buy the solution. This is particularly important due to the marketing value of solutions and the high cost of human resources in Business-to-Business (B2B) scenarios.

While there have been some studies focusing on designing effective matching systems [2, 23, 25, 29, 37, 44, 47], none of these works have explored the matching of cloud solutions and their customers, which holds significant business value. In Huawei Cloud, the scenario is manual-driven, wherein our model identifies a list of the top matching companies to the sales team associated with a specific solution. The sales team then manually reviews this list and proceeds with promoting the solution to those companies. This specific scenario can be considered a matching problem, with the primary goal being the identification of appropriate companies (customers) for the sales teams to target in their promotion efforts.

In this work, we focus on this specific scenario of B2B solution matching and identify **two main challenges**: **(1) The features of solutions and companies are complex and often comprised of multiple fields.** As presented in Table 1, the features consist of text, categorical, and numeric features, which consist of multiple fields. Modeling these different types of features can pose challenges, such as different encoding paradigms for texts and other features, potential interference between different text fields, and interactions between different types of features. **(2) The available transaction data, which include recorded successful purchases, are limited, incomplete, and sparse.** The paradigm of our scenario is manual-driven, *i.e.*, the list of highly matched companies generated by a matching system needs to be manually reviewed by sales teams before contacting potential customers. As a result, the matching of solutions and target companies in a B2B scenario requires significant human resources, leading to limited recorded data. Additionally, the data of solutions and companies may be incomplete, with specific fields or tokens missing due to recording or registration errors (as shown in Table 1). In addition, numerous potential purchases remain undiscovered. For instance, despite similar companies requiring the same solution, they may not be persuaded by the sales team due to various factors, including the effectiveness of the sales pitch, the company's history of purchasing solutions from other providers, and even personal relationships. Consequently, our available training data inherently suffers from data sparsity.

To address these problems, we propose a **C**ontrastive pre-trained hier**A**rchical **M**ulti-field m**A**tching framework for B2B cloud solution matching (CAMA). More specifically:

• **For the first challenge**, we propose a hierarchical multi-field matching framework as the backbone of CAMA. To mitigate token-level interference between different types of texts, we separate the text fields into description-like texts and attribute-like texts. To gain a comprehensive understanding of each text group, we utilize two separate BERT models [11] as token-level encoders. We further incorporate Field-aware Embeddings into the embedding layers of BERTs to identify different fields during the matching process. In addition to the text fields, we also employ look-up embeddings and the AutoDis technique [16] to encode categorical and numerical features, respectively. Furthermore, we model the interactions among different feature groups at a higher level using another Transformer encoder [12]. This allows us to capture the dependencies and relationships between the various feature groups.

• **To address the second challenge**, we devise several data augmentation strategies and implement a contrastive learning objective to pre-train the BERT encoders. To generate additional solution-company sample pairs, we employ three augmentation techniques: Token Masking, Field Masking, and Company Replacing. These strategies allow us to complement the imperfect transaction data by introducing variations. Through pre-training the BERT encoders with augmented data and a contrastive learning objective, we aim to enhance our model's robustness and generalization.

Our experiments in both online and offline settings demonstrate the effectiveness of our proposed model. CAMA performs significantly better than some strong matching baseline models on a real-world dataset. After being deployed on an online system, it surpasses the previous online model of Huawei Cloudby about 30% in terms of Conversion Rate.

In summary, the contributions of our work are as follows:

(1) We recognize the value and importance of the B2B cloud solution matching problem and identify two major challenges.

(2) To address these challenges, we propose CAMA, a framework that incorporates a hierarchical multi-field matching approach and a text-matching enhancement module utilizing contrastive learning.

(3) We validate the effectiveness of our framework through experiments conducted in both offline and online settings. The results demonstrate that our method is effective in the matching of cloud solutions and target companies in real-world B2B scenarios, offering substantial industrial value.

Table 1: An example solution-company pair to present their complex multi-field features. The missing fields and tokens due to recording errors are marked with the color red.

Group	Field	Feature
	Solution	
Description (s^{d})	Name Introduction	Distributed Cache Service There are currently hot data in...
Attribute (s^{a})	First-level Industry Second-level Industry Third-level Industry Target Industry	Internet Internet Information Services NA NA
	Company	
Description (c^{d})	Name Introduction Business Scope	*** (masked for privacy concern) *** is an internet education... The business scope includes...
Attribute (c^{a})	First-level Industry Second-level Industry Third-level Industry Copyright Name Key Project Category	Education (Online) Education Skills training, ... review system, ... Education Platform
Scale (c^{s})	Categorical Features Numeric Features	{Status: 1}, ... (8 more features) {# App: 7}, ... (21 more features)

2 Related Work

2.1 B2B Application Scenario

Business-to-Business (B2B) systems differ from Business-to-Consumer systems typically employed on e-commerce platforms. B2B systems are designed for more complex scenarios and are primarily deployed within a company for internal usage. Consequently, the existing literature on B2B matching is limited. Zhang et al. [49] was an early work introducing B2B matching scenario. Some works combined several techniques to design hybrid systems [1, 31, 36]. For instance, Pande et al. [31] combined case-based reasoning (CBR), Collaborative Filtering (CF), and Random Walk for Consultancy. Furthermore, some applied tree-based algorithm [40] and graph-based methods [18, 41, 42] to model relationships of customers. For example, Henna et al. [18] utilized a graph convolution network (GCN) for B2B customer relationship management. These works all make great contributions to this field. However, none of these studies have explored the valuable solution-company matching scenario and its associated challenges.

2.2 Matching Models

In our scenario, the problem can be considered a matching problem of solutions and companies, *i.e.*, helping salesmen contact the companies that match the solution. In neural text matching, researchers focused on two kinds of models, representation-based [4, 20, 22, 24, 26, 30, 34, 46, 52] and interaction-based [5, 7, 11, 32, 43, 45, 50]. Representation-based models convert sentences into hidden vectors, whereas interaction-based models match texts on word-level

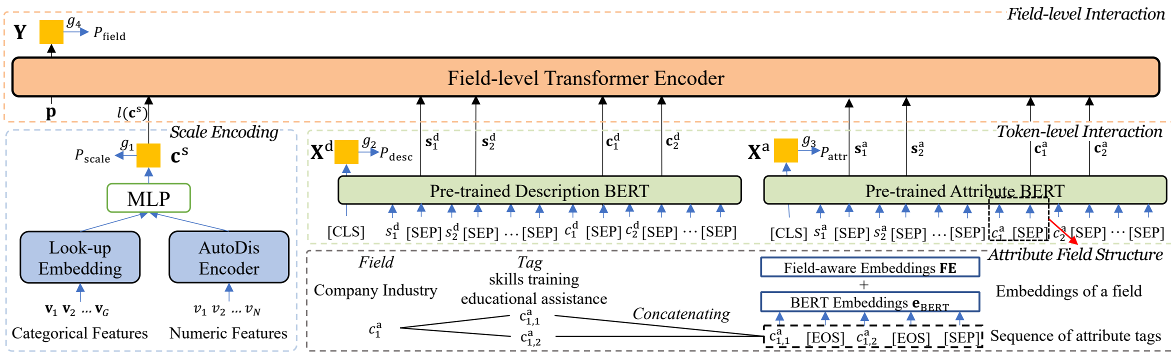

Figure 1: The illustration of CAMA. The scale encoding module incorporates the usage of look-up embedding and the AutoDis encoder to effectively model categorical and numeric features, respectively. Furthermore, two pre-trained BERT encoders are employed along with field-aware embeddings to capture token-level interactions within two distinct groups of text pairs. At a higher level, a Transformer encoder is utilized to model field-level interactions among various feature groups.

interactions. Because of the high cost of human marketing, there is a high requirement for B2B matching systems' accuracy. Thus, we need to model the interactions of the texts.

Some researchers have studied multi-aspect text matching for News Recommendation [13, 39, 48], Document Ranking [47], and Dense Retrieval [2, 23, 27, 37]. For instance, Kong et al. [23] represented multiple aspects of a query using different embeddings. Shan et al. [37] designed an attribute-guided representation learning framework to couple the query and item representation learning together. This framework can also identify the most relevant item features for item representation. To facilitate text matching, we also encode the scale features of companies. Some works have studied to incorporate side information into the text-matching process for Document Ranking [33] and Recommender Systems [8, 17, 35]. Wide&Deep [8] combined a wide linear model and a DNN to capture both sparse features and dense embeddings. Some works of Entity Matching have developed matching frameworks that can be potentially applied to our cloud solution matching problem [25, 29, 44]. For example, HierGAT [44] developed a hierarchical graph attention transformer that utilizes both self-attention and graph-attention mechanisms.

3 Methodology

This study focuses on the matching of solutions and companies in the B2B scenario, which holds significant commercial value but has received limited attention in previous research. To address the challenges of this scenario, we first propose a Hierarchical Multi-field Matching framework to model the complex multi-field features of solutions and companies. Specifically, by considering three aspects of the modeling process, *i.e.*, scale features encoding, fine-grained token-level interaction, and field-level inter-group interaction, we can compute matching scores from different perspectives. Furthermore, recognizing the issue of limited, incomplete, and sparse transaction data, we devise several data augmentation strategies to generate supplemental solution-company data pairs. Additionally, we employ a contrastive learning objective to pre-train our text models, enhancing their ability to learn the intricate interactions between solutions and companies.

3.1 Problem Definition

Before shedding light on our model, we first give a concise definition of the problem we study. Our objective is to identify potential buyers (companies) to the sales teams based on the outcomes of our framework. Specifically, we denote the solutions as S and the companies as C. For each solution $s \in S$, we need to rank C based on the matching scores between s and every company $c \in C$, denoted as $P(s, c)$. As presented in Table 1, the fields of s and c are divided into three groups: description texts (s^{d}, c^{d}), attribute texts (s^{a}, c^{a}), and scale features (c^{s}) (only c has categorical and numerical features representing its scale). We categorize the text features into two distinct groups based on their meanings and structures: (1) The description texts are typically long natural language texts consisting of general descriptions of s and c. On the other hand, attribute texts include keywords or tags that represent attributes of s and c. Due to the distinct nature of these two types of texts, their token-level interactions can interfere with each other (demonstrated in Section 5.2). (2) The text features of solutions and companies are heterogeneous. In other words, the features of s and c are not exact matches. Thus, it is not feasible to treat each field as a group and perform field-to-field matching between s and c.

The top-ranked company list of C will be distributed to the sales team responsible for promoting the corresponding solution, who will subsequently contact these companies and pursue potential sales opportunities. By leveraging the multi-field features, the model aims to learn the patterns of matching between s and c and subsequently rank the companies most likely to purchase s as high as possible within the generated lists.

3.2 Framework Overview

In this part, we will briefly introduce the overall structure of our framework. Our framework is comprised of two parts:

(1) **Hierarchical Multi-field Matching.** As shown in the lower part of Fig. 1, we design a hierarchical matching structure to effectively capture the interactions between solutions and companies whose features are comprised of complex fields. To begin with, we focus on capturing the fine-grained token-level interactions within two groups of text fields. This is achieved by utilizing BERT encoders and field-aware embeddings, which allows us to extract rich representations from the textual data. We also encode scale features of c into a representation that captures the company's scale. Subsequently, we employ a Transformer encoder to model the field-level inter-group interactions. Four matching scores are calculated based on the interactions from different perspectives.

(2) **Text Matching Enhancement.** In this part, we attempt to enhance the token-level interactions of the text features. To achieve this, we design three data augmentation strategies and a contrastive objective to pre-train the BERT encoders. Specifically, for each (s, c) pair in the training data, we randomly select two strategies to generate two similar pairs. Subsequently, we apply a contrastive loss function to pull together the representations of the generated pairs and push them away from other pairs within the same mini-batch. By pre-training our BERT encoders with this objective, we effectively improve the modeling of text interaction, especially with limited, incomplete, and sparse data.

3.3 Hierarchical Multi-field Matching

In our B2B cloud solution matching scenario, we have identified a significant and unexplored challenge: the modeling of complex multi-field feature interactions. Specifically, the fields of solutions and companies are comprised of two main kinds of features: scale features and text features. Consequently, we devise distinct models to effectively capture and analyze these different types of features.

3.3.1 Scale Encoding. Since only c has scale features that represent its scale, our goal here is to encode these features into a representation rather than modeling interactions. Instead of using c^{s} directly, we encode it to better interact with the textual representations. Besides, c^{s} is comprised of both categorical (*e.g.*, whether c is listed) and numerical (*e.g.*, registration capital) fields. Therefore, we encode these two types of features separately and fuse them into a unified representation, as illustrated in the lower left part of Fig. 1.

Suppose c^{s} contains G categorical fields and N numerical fields: $c^{\mathrm{s}} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_G; v_1, v_2, \ldots, v_N]$, where $\mathbf{v}_i$ is the one-hot vector of the value of the i-th categorical field, and v_j is the scalar value of the j-th field of the numerical features.

For the categorical fields, we apply a look-up embedding technique to encode the one-hot vectors. Specifically, for the i-th categorical field, we obtain its embedding: $\mathbf{e}_i = \mathbf{E}_i \cdot \mathbf{v}_i$, where $\mathbf{E}_i \in \mathbb{R}^{f_i \times d_{\mathrm{s}}}$ is the embedding matrix for the i-th field to look-up, f_i is the field size, and d_{s} is the embedding size of all scale features.

To handle the numerical fields, we employ an automatic embedding learning technique based on soft discretization (AutoDis [16]). The utilization of soft discretization within our end-to-end learning framework allows for the optimization of this process. First, the scalar value v_j is discretized into buckets by a two-layer neural network with skip-connection $\mathbf{h}_j = \text{Leaky\_ReLU}(\mathbf{w}_j v_j), \widetilde{\mathbf{v}}_j = \mathbf{W}_j \mathbf{h}_j + \alpha \mathbf{h}_j$, where $\mathbf{w}_j \in \mathbb{R}^{1 \times H_j}$ and $\mathbf{W}_j \in \mathbb{R}^{H_j \times H_j}$ are learnable parameters that automatically discretized v_j into the projection outputs of H_j buckets: $\widetilde{\mathbf{v}}_j = [\widetilde{v}_j^1, \widetilde{v}_j^2, \ldots, \widetilde{v}_j^{H_j}]$, where $\widetilde{v}_j^h$ is the projection of scalar value v_j on the h-th bucket. This projection is then normalized by the Softmax$(\cdot)$ function into a weight on the corresponding bucket: $\widehat{v}_j^h = \text{Softmax}(\widetilde{v}_j^h)$.

Subsequently, a set of meta-embeddings $\mathbf{ME}_j \in \mathbb{R}^{H_j \times d_{\mathrm{s}}}$ is designed for the j-th field. The softly discretized results $\widehat{\mathbf{v}}_j$ represents the relevance between the j-th field of the numeric features and the buckets of meta-embeddings. Thus, we can leverage a weighted-average technique to aggregate the meta-embeddings and their corresponding weights into a representation for each numerical feature: $\mathbf{e}_j^{\mathrm{num}} = \sum_{h=1}^{H_j} \widehat{v}_j^h \cdot \mathbf{ME}_j^h$.

After the scale features are embedded into continuous vectors, we employ a Multi-Layer Perceptron (MLP) to fuse them into a unified representation $\mathbf{c}^{\mathrm{s}} \in \mathbb{R}^{d_{\mathrm{s}}}$ that captures the scale features of c: $\mathbf{c}^{\mathrm{s}} = \text{MLP}\big([\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_G; \mathbf{e}_1^{\mathrm{num}}, \mathbf{e}_2^{\mathrm{num}}, \ldots, \mathbf{e}_N^{\mathrm{num}}]\big)$. We can get a score by applying a linear projection $g_1(\cdot)$ to map this representation into a (scalar) score: $P_{\mathrm{scale}}(s, c) = g_1(\mathbf{c}^{\mathrm{s}})$.

3.3.2 Token-level Interaction. In this part, we attempt to model the token-level interactions of (s, c). Pre-trained language models, such as BERT [11], have gained significant popularity in various tasks, including Recommender Systems [19, 38] and Information Retrieval [6, 51]. To capture the fine-grained token-level interactions of $(s^{\mathrm{d}}, c^{\mathrm{d}})$ and $(s^{\mathrm{a}}, c^{\mathrm{a}})$, we leverage BERT as the underlying encoder. We employ special tokens to concatenate the fields in the description texts, resulting in the following sequence:

$$X^{\mathrm{d}} = [\mathrm{CLS}] s_1^{\mathrm{d}} [\mathrm{SEP}] \ldots s_{F_{\mathrm{sd}}}^{\mathrm{d}} [\mathrm{SEP}][\mathrm{SEP}] c_1^{\mathrm{d}} \ldots c_{F_{\mathrm{cd}}}^{\mathrm{d}} [\mathrm{SEP}][\mathrm{SEP}],$$

where F is the number of text fields, s_i and c_j are the i-th and j-th fields that consist of many tokens of the solution and the company, respectively, "[CLS]" is the token used for representing the sequence, "[SEP]" is the separator token. We append a "[SEP]" token after each field to indicate the end of a field and another one at the end of each sequence of fields. Moreover, to distinguish the multiple tags in attribute texts (*e.g.*, different industry tags), we further utilize "[EOS]" tokens to separate these tags before concatenating them (as shown in the lower center part of Fig. 1):

$$c_j^{\mathrm{a}} = c_{j,1}^{\mathrm{a}} [\mathrm{EOS}] c_{j,2}^{\mathrm{a}} [\mathrm{EOS}] \ldots c_{j,T_j}^{\mathrm{a}} [\mathrm{EOS}],$$
$$X^{\mathrm{a}} = [\mathrm{CLS}] s_1^{\mathrm{a}} [\mathrm{SEP}] \ldots s_{F_{\mathrm{sa}}}^{\mathrm{a}} [\mathrm{SEP}][\mathrm{SEP}] c_1^{\mathrm{a}} \ldots c_{F_{\mathrm{ca}}}^{\mathrm{a}} [\mathrm{SEP}][\mathrm{SEP}],$$

where c_j^{a} is the j-th attribute field of the company and T_j is the number of the tags contained in this field.

Furthermore, we design a set of field-aware embeddings that help the encoders to distinguish different fields during the modeling of interactions. Specifically, for each text group, we initialize a field-aware embedding matrix $\mathbf{FE} \in \mathbb{R}^{F \times d_{\mathrm{e}}}$, where F is the number of the fields and d_{e} is the size of BERT's word embeddings. The "[EOS]" tokens within and the "[SEP]" tokens after fields are also enhanced with corresponding field-aware embeddings (the **FE** of "[CLS]" is distinct from others). Consequently, the embedding of each token is comprised of both our field-aware embedding and BERT embedding (as shown in the lower right part of Fig. 1): $\mathbf{X}^{\mathrm{d}} = \mathrm{BERT}_{[\mathrm{CLS}]}^{\mathrm{d}}\big(\mathbf{FE}^{\mathrm{d}} + \mathbf{e}_{\mathrm{BERT}}(X^{\mathrm{d}})\big), \mathbf{X}^{\mathrm{a}} = \mathrm{BERT}_{[\mathrm{CLS}]}^{\mathrm{a}}\big(\mathbf{FE}^{\mathrm{a}} + \mathbf{e}_{\mathrm{BERT}}(X^{\mathrm{a}})\big)$, where $\mathbf{e}_{\mathrm{BERT}}(\cdot)$ is the embedding layer of BERT.

We can get two token-level matching scores by applying linear projections $g_2(\cdot)$ and $g_3(\cdot)$ on $\mathbf{X}^{\mathrm{d}}$ and $\mathbf{X}^{\mathrm{a}}$, respectively: $P_{\mathrm{desc}}(s, c) = g_2(\mathbf{X}^{\mathrm{d}})$, $P_{\mathrm{attr}}(s, c) = g_3(\mathbf{X}^{\mathrm{a}})$.

3.3.3 Field-level Interaction. In the previous section, we have gathered information regarding token-level interactions within two distinct text groups. However, although it has been observed that token-level interactions between different groups can have a detrimental effect on model performance, we still aim to capture the inter-group interactions from a higher level, *i.e.*, the field level. As shown in the center part of Fig. 1, for each text field, we use the encoded output of the "[SEP]" token appended to it as its representation. By doing so, we avoid using all tokens for the same reason we divide the features into two groups, *i.e.*, preventing fine-grained interference. Additionally, we use the encoded scale representation $\mathbf{c}^{\mathrm{s}}$ to facilitate the modeling of field-level interactions. This is because the scale of a company can influence how a solution interacts with it. For instance, interactions with smaller companies may place more emphasis on their copyright works due to their more focused business nature.

In order to comprehensively model these representations and capture the interactions among fields, we utilize the Transformer encoder, as proposed in the Transformer [12] architecture. The Transformer encoder effectively models the aforementioned representations in the following manner:

$$\begin{aligned}\mathbf{Y} = \mathrm{Trm}_{\mathbf{p}}\big([\mathbf{p}; l(\mathbf{c}^{\mathrm{s}}); \mathbf{s}_1^{\mathrm{d}}, \dots, \mathbf{s}_{F_{\mathrm{sd}}}^{\mathrm{d}}, \mathbf{c}_1^{\mathrm{d}}, \dots, \mathbf{c}_{F_{\mathrm{cd}}}^{\mathrm{d}};\\ \mathbf{s}_1^{\mathrm{a}}, \dots, \mathbf{s}_{F_{\mathrm{sa}}}^{\mathrm{a}}, \mathbf{c}_1^{\mathrm{a}}, \dots, \mathbf{c}_{F_{\mathrm{ca}}}^{\mathrm{a}}]\big),\end{aligned} \tag{1}$$

where $\mathrm{Trm}(\cdot)$ is the Transformer encoder which consists of k Transformer layers, $\mathbf{s}$ and $\mathbf{c}$ are the encoded outputs of "[SEP]" appended to text fields, $l(\cdot)$ is a linear projector to map $\mathbf{c}^{\mathrm{s}}$ into the latent space of the text field representations, and $\mathbf{p} \in \mathbb{R}^{d_{\mathrm{e}}}$ is a randomly initialized vector used for pooling.

We can get a field-level matching score by applying a linear projection $g_4(\cdot)$: $P_{\mathrm{field}}(s, c) = g_4(\mathbf{Y})$.

3.3.4 Optimization. Since the label of whether a company buys a solution is binary (0/1), we use a cross-entropy loss to optimize our matching model. We formulate the loss for the matching scores of our hierarchical models, *i.e.*, $\{\mathcal{P}\} = \{P_{\mathrm{scale}}, P_{\mathrm{desc}}, P_{\mathrm{attr}}, P_{\mathrm{field}}\}$:

$$\mathcal{L}_{\mathrm{Match}} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{P\in\{\mathcal{P}\}} y_i \log P_i + (1 - y_i)\log(1 - P_i), \tag{2}$$

where M is the number of (s, c) pairs in the training set, and y_i is the label of the i-th data pair.

3.4 Text Matching Enhancement

In our study, we identify several challenges of the transaction data in our scenario. Firstly, the training data is limited attributed to the high cost of human resources required for promoting solutions. Additionally, the data is incomplete as some solutions and companies lack specific tokens or fields. Moreover, the data is sparse, meaning that many potential pairs have not been discovered and recorded. To mitigate these challenges, we attempt to enhance the generalization and robustness of the BERT encoders by pre-training them. As shown in Fig. 2, we employ a contrastive objective, which aims to bring together the representations of augmented similar sequences while pushing away different ones.

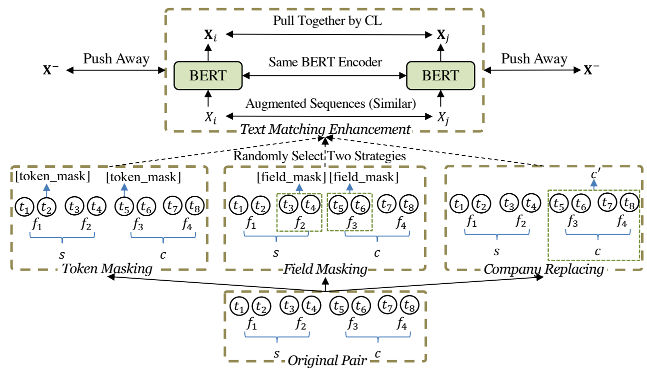

Figure 2: The illustration of our data augmentation strategies and contrastive learning process. Initially, an original (s, c) pair is augmented by two random strategies. The BERT-encoded representations of these two similar pairs are then brought closer through our contrastive loss function.

3.4.1 Data Augmentation Strategies. The similar sequences are generated from the inputs of BERT encoders (X). We design three data augmentation strategies to generate additional data to complement the limited transaction data in our scenario.

(1) **Token Masking.** Some existing works in Natural Language Processing [3, 9, 14] have employed token-level augmentation techniques to enhance the robustness of sentence representations. This approach enables our BERT encoders to acquire more robust and generalized representations with incomplete data that may lack specific tokens, by reducing reliance on specific tokens.

To begin, we represent the token-level input sequence X as a token sequence: $X = [t_1, t_2, \dots, t_W]$, where W denotes the total number of tokens. Next, we randomly mask a proportion r_{t} of the tokens in X: $T_{\mathrm{m}} = [t'_1, t'_2, \dots, t'_{M_{\mathrm{T}}}]$, where t'_i is the token to be masked and $M_{\mathrm{T}} = \lfloor r_{\mathrm{t}} * W \rfloor$. For each token $t_i \in X$, if it is marked to be masked in T_{m}, we will replace it with a special token "[token_mask]", which is similar to "[MASK]" in BERT [11].

(2) **Field Masking.** To obtain robust representations of incomplete data pairs, our BERT encoder models should avoid relying on specific data fields while encoding the whole matching sequences. By contrasting sequences that are augmented through the masking of certain fields with other sequences, we can facilitate the learning of text matching for incomplete data within our BERT models.

We represent X as a sequence of data fields: $X = [f_1, f_2, \dots, f_F]$, where F is the total number of fields. We randomly mask a ratio r_{f} of the fields: $D_{\mathrm{m}} = [f'_1, f'_2, \dots, f'_{F_{\mathrm{T}}}]$, where f'_i represents the token to be masked and $F_{\mathrm{T}} = \lfloor r_{\mathrm{f}} * F \rfloor$. For $f_i \in X$, if it is marked to be masked in D_{m}, we will replace it with a special token "[field_mask]".

(3) **Company Replacing.** To address the data sparsity problem, we design a rule-based method to identify similar companies[1] that can replace the company in the positive pairs. Specifically, for each

[1]We only find similar companies because the number of companies is much larger than the number of solutions, and solutions are often distinctive.

sequence X of positive pairs (s, c), we find one similar company c' to generate a pair (s, c'). Because the company name serves as the primary identifier, capturing information about the industry and business nature, we only use the name to identify similar companies. This simplification aims to expedite computations and diminish the complexity of the rule. Our rule-based method randomly selects one of the five most similar companies of company c to be designated as company c'. The similarity between companies is obtained by calculating the semantic similarity of company names using the sentence-transformers package [34].

3.4.2 Contrastive Learning Objective. Inspired by several studies in Information Retrieval [10, 15, 21, 28, 51], we adopt a contrastive learning approach to enhance our model for text matching. In order to carry out contrastive learning for our task, we define a loss function specifically for the contrastive prediction task. This task involves identifying similar augmented pairs within the set $\{\mathcal{X}\}$, which is constructed by randomly augmenting the original sequences of a minibatch. Suppose a minibatch contains M sequences, we randomly employ two of our augmentation strategies to generate $\{\mathcal{X}\}$ comprising $2M$ sequences. Among these sequences, the two sequences generated from the same solution-company pair are considered a similar (positive) pair, whereas the remaining $2(M-1)$ augmented ones serve as negative samples $\{\mathcal{X}\}^-$. We formulate the contrastive learning loss for a positive pair $(\mathbf{X}_i, \mathbf{X}_j)$ as follows:

$$\phi(\cdot) = \exp\left(\frac{\text{Sim}(\cdot)}{\tau}\right), \tag{3}$$

$$\mathcal{L}_{\text{CL}}(i, j) = -\log \frac{\phi(\mathbf{X}_i, \mathbf{X}_j)}{\phi(\mathbf{X}_i, \mathbf{X}_j) + \sum_{\mathbf{X}^- \in \{\mathcal{X}\}^-} \phi(\mathbf{X}_i, \mathbf{X}^-)}, \tag{4}$$

where $\text{Sim}(\cdot)$ is a similarity function which is defined as cosine similarity in our study, and τ is a hyperparameter temperature.

4 Experimental Setup

4.1 Dataset

To evaluate the effectiveness of our proposed method, we conducted extensive experiments on a real-world dataset. We first sampled from the real-world data and constructed an offline dataset *B2B Solution Matching* (BSM). BSM contains three parts: solution data, company data, and transaction data. The solution dataset stores the text information of 27 solutions, such as solution names, detailed descriptions, and tags of their industries. The company dataset provides detailed company profiles for 533,784 companies, including company names, company registration capital, and their business scopes, etc. The transaction data are derived from the marketing feedback of the sales teams online. Whether a company buys a solution is relevant to not only its suitability but also many subjective factors (*e.g.*, the sales team's pitch). Consequently, unsuccessful purchases are often noise and disregarded.

We constructed the data format as a pairwise form (solution, company) along with their corresponding label from the abovementioned raw dataset. Then the constructed dataset is split into three parts, where 70% is used for training, 10% is used for validating and the remaining 20% is for testing. For each positive sample, we randomly sampled 4 negative companies from the company dataset. The statistics of our dataset are presented in Table 2.

Table 2: Statistics of the BSM dataset.

Datasets	Training	Validation	Testing
# total samples	48,515	6,930	13,860
# positive samples	9,703	1,386	2,772
# negative samples	38,812	5,544	11,088

4.2 Evaluation Metrics

We comprehensively evaluate our proposed method through both offline and online evaluations.

(1) **Offline Metrics.** We use Mean Average Precision (MAP), Area Under Curve (AUC), Precision at k (P@k, k = 10, 100, 500), and Recall at position k (R@k, k = 10, 100, 500) as offline metrics. The results are calculated by averaging across different solutions.

(2) **Online Metrics.** We verify the performance of our proposed model with *Conversion Rate (CVR)* as online metrics. CVR is defined as: CVR = # Purchase/# Promoted, # Purchase denotes the number of companies that purchase the solutions and # Promoted means the number of companies that the sales teams market their solutions. To evaluate CAMA's effectiveness, we will compare its performance to a previous online model of Huawei.

4.3 Baseline Models

We compare our CAMA with two kinds of baselines:

(1) **Text Matching Models** only use the interactions of texts to get the matching score. • Sentence-BERT (SBERT) [34] encodes the solution and company with two BERTs separately and takes the cosine similarity of these representations as the matching score. • MADR [23] is also a representation-based model that designs aspect learning tasks to extract representations of different aspects from texts and then fuses them by calculating the weighted sum. • Concatenating-BERT (CBERT) is an interaction-based model that models the token-level interaction of the solution and company without identifying different text fields.

(2) **Side-aware Matching Models** have frameworks that can encode side information (scale) along with the text interaction module to get the representations for matching. • Wide & Deep (W&D) [8] combines a wide linear model and a DNN. For its implementation, we utilize a two-tower BERT model to get text representations and fit them into the framework with scale features. • MCN [33] conducts an element-wise matching of solutions and companies, and fuses the matching output with scale features to learn interactive information. For the encoding modules in MCN, we use a two-tower BERT to get the textual representations and our scale encoding module to model scale features. We also apply some entity matching frameworks to our solution matching problem by treating our text and numeric fields as different entities. • DeepMatcher (DM) [29] uses the GRU-RNN model to learn the attribute embeddings of entities, which are aggregated for matching. • Ditto [25] applies pre-trained Transformer-based language models and optimization techniques to perform the sequence-pair classification

Table 3: Overall results on BSM. "†" denotes our model outperforms all baselines significantly in paired t-test at $p < 0.01$ level (with Bonferroni correction). The best performance is in bold and the second-best performance is underlined.

	SBERT	MADR	CBERT	DM	W&D	HSCM	Ditto	MCN	HieGAT	CAMA
MAP	0.1953	0.2259	0.2539	0.4084	0.4178	0.4534	0.4711	0.5392	<u>0.5799</u>	**0.6810**†
AUC	0.7135	0.7238	0.7721	0.8093	0.8141	0.8136	0.8206	0.8451	<u>0.8480</u>	**0.8528**†
Rec@10	0.1809	0.2187	0.3283	0.3468	0.3579	0.3740	0.3887	0.4176	<u>0.4361</u>	**0.4579**†
Rec@100	0.2698	0.3685	0.3947	0.4782	0.4947	0.5396	0.5570	0.6295	<u>0.6727</u>	**0.7394**†
Rec@500	0.4156	0.5731	0.5581	0.7211	0.7372	0.7503	0.7773	0.8035	<u>0.8783</u>	**0.9335**†
Pre@10	0.3679	0.4084	0.4723	0.5021	0.5183	0.5420	0.5682	0.6168	<u>0.6492</u>	**0.6807**†
Pre@100	0.2561	0.2795	0.2810	0.2988	0.3019	0.3121	0.3344	0.3642	<u>0.3973</u>	**0.4266**†
Pre@500	0.0620	0.1068	0.1034	0.1529	0.1551	0.1634	0.1659	0.1740	<u>0.2291</u>	**0.2688**†

problem. • HierGAT [44] combines Transformer attention with hierarchical graph attention to effectively learn entity embeddings.

(3) Hybrid Solution-Company Matching Framework (HSCM). This is a previously online yet unpublished framework used by Huawei Cloud. Based on the number of solutions in transaction data, the algorithm categorizes solutions into three groups. It devises unsupervised, semi-supervised, and supervised models for each category, respectively. It is a well-designed classification framework based on the classical Gradient Boosting Decision Tree (GBDT) model and BERT-encoded features. Due to the company's confidentiality policy, it is not feasible for us to provide further elaboration. However, HCSM has the following drawbacks: (1) The framework maintains a distinct model for each solution, leading to a significant waste of resources. (2) Because of its naive structure, the results of the matching are unsatisfactory.

4.4 Implementation Details

We use BERT provided by Huggingface as the token-level encoder[2]. The size of scale embedding d_s is set as 64, and the parameters of AutoDis encoder are the same as its original paper [16]. We use $k = 6$ Transformer layers as the field-level encoder. During the text matching enhancement process, we set the temperatures as 0.2 and 0.05 for pre-training description BERT (τ_d) and attribute BERT (τ_a), respectively. The token mask ratio and field mask ratio are tuned and established as 0.2 and 0.5, respectively, for both BERTs. The learning rates are set as 3e-5 for the token-level encoding module, 5e-4 for the scale encoding module, and 5e-5 for both field-level encoding and pre-training. The model is trained with a batch size of 32 for four epochs using four Tesla P100 16G GPUs.

5 Results and Analysis

To compare our proposed model CAMA with baselines, we perform experiments in both offline and online settings. We primarily conduct detailed experiments on BSM offline. We then deploy CAMA on the online system to observe its overall performance.

5.1 Overall Results

5.1.1 Offline Results. Experimental results on BSM are presented in Table 3. The results show that the performance of CAMA is significantly better than baseline models. We can make the following observations based on the results:

(1) Our proposed method CAMA outperforms all baselines. The offline results demonstrate that CAMA performs significantly better than all baseline models. This indicates that our hierarchical multi-field matching framework and contrastive pre-training technique are effective for matching solutions and companies.

(2) Side-aware models generally perform better than the models purely based on text matching. For example, the weak side-aware model W&D still outperforms the strong text-matching model CBERT. This demonstrates the necessity of modeling the scale features. Moreover, our model still performs better than all side-aware baselines, which demonstrates the effectiveness of our hierarchical multi-field matching framework again.

5.1.2 Online Deployment Results. To assess our proposed model, CAMA, we implemented it in a real-world system over a six-month period. Our method, when given a solution slated for sale, generates a Top-K ranked list. The sales team for the solution then reviews this list, selecting companies based on the provided matching scores and their own expertise. Following this, salesmen contact the chosen companies and gather feedback to compute the Conversion Rate (CVR). Due to confidentiality constraints, we cannot reveal specific CVR figures. However, we present the relative CVR increase ratios during the evaluation period. Our approach involved having the same sales teams process both the company lists generated by our model and those produced by the pre-online model HSCM. By comparing the CVR results from these two models, CAMA demonstrated a 29.99% improvement over the HSCM model during the same timeframe. Moreover, we can make these observations:

(1) Our proposed method CAMA achieves a superior performance over HSCM. The performance of CAMA surpasses the previous online model HSCM by 29.99%, which demonstrates its great commercial value.

(2) Our proposed method CAMA is not only better than the previous framework HSCM but also easy to deploy and maintain. The online deployment results validate the effectiveness of the hierarchical multi-field matching structure and the contrastive pre-trained method compared with the improvement over HSCM.

[2]https://huggingface.co/hfl/chinese-bert-wwm-ext

Table 4: Performances of ablated models on BSM.

Metric	MAP		Rec@10	
CAMA (Full)	**0.6810**	-	**0.4579**	-
w/o. Description Texts	0.4891	-28.18%	0.3905	-14.72%
w/o. Attribute Texts	0.5885	-13.59%	0.4243	-7.33%
w/o. Text Grouping	0.4866	-28.55%	0.3888	-15.09%
w/o. Field Embeddings	0.5927	-12.97%	0.4266	-6.83%
w/o. Scale Encoding	0.5596	-17.83%	0.4162	-9.11%
w/o. Field-level Interaction	0.6386	-6.23%	0.4393	-4.07%
w/o. Pre-training	0.5919	-13.09%	0.4259	-6.98%

Furthermore, CAMA provides a unified model for all solutions which makes it easy to maintain in online deployment.

5.2 Ablation Studies

To evaluate the effectiveness of each component, we perform the following ablation studies on the BSM dataset:

• **CAMA w/o. Description Texts** is CAMA without the interactions of Description Texts, *i.e.*, without P_{desc} and the corresponding field-level representations. • **CAMA w/o. Attribute Texts** is CAMA without the interactions of Attribute Texts. • **CAMA w/o. Text Grouping** is CAMA without Text Grouping, *i.e.*, we do not distinguish two types of texts and only compute one matching score ($\{P_{\text{desc}}, P_{\text{attr}}\} \rightarrow P_{\text{text}}$). • **CAMA w/o. Field Embeddings** is CAMA without the Field-aware Embeddings. • **CAMA w/o. Scale Encoding** is CAMA without the Scale Encoding component. • **CAMA w/o. Field-level Interaction** is CAMA without the Field-level Interaction module, *i.e.*, P_{field}. • **CAMA w/o. Pre-training** is CAMA without the pre-training of the token-level BERT encoders.

The results in Table 4 clearly demonstrate that the full model outperforms all ablated models, indicating the effectiveness of our components. Moreover, we can make the following conclusions:

(1) Both the interactions of two text groups can help the matching process. Specifically, CAMA's performance decreases by about 28.18% and 13.59% in terms of MAP after abandoning Description and Attribute Texts, respectively. This indicates the importance of modeling both groups of texts.

(2) The token-level interactions between two text groups interfere with each other. Our model's performance decreases by about 15.09% in terms of Rec@10 after we combine two text groups and do not distinguish them in the token-level interaction module.

(3) Identifying different text fields with field-aware embeddings is effective. The decrease of CAMA after discarding Field-aware Embeddings demonstrates its effectiveness.

(4) Modeling the scale of companies can facilitate our hierarchical textual interactions. This decrease of CAMA after abandoning Scale Encoding meets our observation in Section 5.1, that company scale plays an important role in modeling the matching between solutions and companies.

(5) Field-level interaction is effective for capturing intergroup matching signals. The performance of CAMA decreases after abandoning the high-level interaction among fields of different groups. This indicates the effectiveness of our field-level interaction.

Table 5: Performances of CAMA without different data augmentation strategies on BSM.

Metric	MAP		Rec@10	
CAMA (Full)	**0.6810**	-	**0.4579**	-
w/o. Token Masking	0.6479	-4.86%	0.4463	-2.54%
w/o. Field Masking	0.6097	-10.47%	0.4326	-5.53%
w/o. Company Replacing	0.5961	-12.47%	0.4274	-6.65%

(6) Utilizing data augmentations and contrastive learning to pre-train our BERT encoders can make CAMA more generalized. Specifically, our model's performance decreases by about 13.09% in terms of MAP. This validates that our data augmentation strategies and contrastive objective can make CAMA more robust.

5.3 Influence of Data Augmentation Strategies

In our text matching enhancement module, we propose three data augmentation strategies and a contrastive learning objective to pre-train the BERT encoders. To investigate the influence of these strategies, we compare CAMA with the ablated models and show the results in Table 5. We can make the following observations:

(1) Token masking strategy can help pre-train the BERT encoders. After abandoning the sequences generated by masking specific tokens, the performance of our model drops by about 4.86% in terms of MAP. This shows that the available transaction data are deficient for training a robust model, and our token masking strategy can help CAMA get more generalized representations.

(2) Pairs augmented by field masking can mitigate the problem of incomplete data. There are some solutions and companies lack specific data fields. The performance of CAMA w/o. Field Masking validates this problem of our data and demonstrates that our strategy can help mitigate this challenge.

(3) Replacing the company in a matching pair with a similar company can generate data pairs for addressing data sparsity. The performance of our model decreases by about 6.65% in terms of Rec@10 after discarding Company Replacing, which demonstrates this technique can complement the sparse data.

5.4 Influence of Hyperparameters

5.4.1 The Embedding Size of Company Scale. To model the interactions between scale features and text embeddings, we have to embed the scale features into a vector $\mathbf{c}^{\text{s}} \in \mathbb{R}^{d_{\text{s}}}$. We tune the scale embedding size d_{s} in the range $[16, 256]$ with the step of 16 on the validation set and present the performances of CAMA with different d_{s} on the test set. Due to the space limitation, we present the performance of MAP on BSM and only show the results of five tuned values (the henceforth displays will follow the same policy). As shown in the left part of Fig. 3, the performances increase to the optimal value and then drop. This pattern indicates a trade-off: If the embedding size is too low, the scale embedding can not encode sufficient information. However, it may introduce noise into CAMA if the embedding size is too high.

5.4.2 The Temperature of Contrastive Learning. In our contrastive loss, there is a hyperparameter τ representing the temperature hyperparameter which controls the model's discrimination against

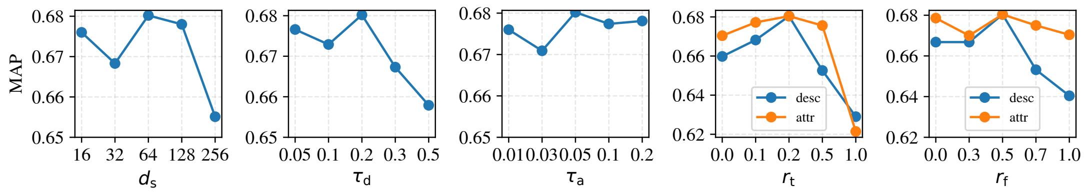

Figure 3: Performance of CAMA on the BSM dataset with different hyperparameters.

negative samples. If it is set too low, our model will concentrate on the negative pairs that are hard to distinguish. However, a high value of τ will make CAMA treat all negative samples equally. We tune τ_d in the range $[0.05, 0.5]$ with the step of 0.05 and τ_a in the range $[0.01, 0.2]$ with the step of 0.01. The results presented in the middle part of Fig. 3 demonstrate the trade-off of this hyperparameter and validate our choices.

5.4.3 The Mask Ratios. In our data augmentation strategies, we use mask ratios r_t and r_f to control the number of tokens/fields we mask in Token Masking and Field Masking strategies, respectively. We tune these two ratios for pre-training two BERTs and find that the patterns of the two BERTs are the same, thus we present the tuning results of each ratio for both BERTs in the same figure. Specifically, we tune r_t and r_f in the range $[0.0, 1.0]$ with the step of 0.05. The results shown in the right part of Fig. 3 indicate there is also a trade-off for the masking ratios. If we set the masking ratios too high, the augmented pairs may not be similar to the original pair. However, too low masking ratios will introduce little knowledge into our pre-training process, resulting in insufficient pre-training.

6 Conclusion

In this work, we study a valuable yet understudied problem of B2B solution matching and identify two key challenges in this scenario. Initially, we propose a hierarchical multi-field matching framework to model the interactions between the complex multi-field features of solutions and companies. Subsequently, three data augmentation strategies and a contrastive learning objective are proposed to deal with the limited, incomplete, and sparse transaction data. Extensive experiments on a real-world dataset BSM demonstrate the effectiveness of CAMA. The deployment of our framework on Huawei Cloud validates the feasibility and effectiveness of our framework in an online scenario. Considering the generalizability of our framework, it can also be applied to other B2B matching scenarios that encounter similar challenges.

Acknowledgments

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 62272467, the fund for building world-class universities (disciplines) of Renmin University of China, and Public Computing Cloud, Renmin University of China. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE, and Beijing Key Laboratory of Big Data Management and Analysis Methods.

References

[1] Sundus Ayyaz, Usman Qamar, and Raheel Nawaz. 2018. HCF-CRS: A Hybrid Content based Fuzzy Conformal Recommender System for providing recommendations with confidence. *PloS one* 13, 10 (2018), e0204849.

[2] Saeid Balaneshinkordan, Alexander Kotov, and Fedor Nikolaev. 2018. Attentive Neural Architecture for Ad-hoc Structured Document Retrieval. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 1173–1182. https://doi.org/10.1145/3269206.3271801

[3] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, Yoav Goldberg and Stefan Riezler (Eds.). ACL, 10–21. https://doi.org/10.18653/v1/k16-1002

[4] Haonan Chen, Zhicheng Dou, Kelong Mao, Jiongnan Liu, and Ziliang Zhao. 2024. Generalizing Conversational Dense Retrieval via LLM-Cognition Data Augmentation. *CoRR* abs/2402.07092 (2024). https://doi.org/10.48550/ARXIV.2402.07092 arXiv:2402.07092

[5] Haonan Chen, Zhicheng Dou*, Qiannan Zhu, Xiaochen Zuo, and Ji-Rong Wen. 2022. Integrating Representation and Interaction for Context-Aware Document Ranking. *ACM Trans. Inf. Syst.* (2022). https://doi.org/10.1145/3529955

[6] Haonan Chen, Zhicheng Dou, Yutao Zhu, Zhao Cao, Xiaohua Cheng, and Ji-Rong Wen. 2022. Enhancing User Behavior Sequence Modeling by Generative Tasks for Session Search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 180–190. https://doi.org/10.1145/3511808.3557310

[7] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 1657–1668. https://doi.org/10.18653/v1/P17-1152

[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) *(DLRS 2016)*. Association for Computing Machinery, New York, NY, USA, 7–10. https://doi.org/10.1145/2988450.2988454

[9] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 3079–3087. https://proceedings.neurips.cc/paper/2015/hash/7137debd45ae4d0ab9aa\953017286b20-Abstract.html

[10] Shitong Dai, Jiongnan Liu, Zhicheng Dou, Haonan Wang, Lin Liu, Bo Long, and Ji-Rong Wen. 2023. Contrastive Learning for User Sequence Representation in Personalized Product Search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, 380–389. https://doi.org/10.1145/3580305.3599287

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[12] Markus Eisenbach, Jannik Lübberstedt, Dustin Aganian, and Horst-Michael Gross. 2023. A Little Bit Attention Is All You Need for Person Re-Identification. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*. IEEE, 7598–7605. https://doi.org/10.1109/ICRA48891.2023.

10160304
[13] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 278–288. https://doi.org/10.1145/2736277.2741667
[14] Shubhashri G, Unnamalai N, and Kamalika G. 2018. LAWBO: a smart lawyer chatbot. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2018, Goa, India, January 11-13, 2018*. ACM, 348–351. https://doi.org/10.1145/3152494.3167988
[15] Leilei Gan, Baokui Li, Kun Kuang, Yi Yang, and Fei Wu. 2022. Exploiting Contrastive Learning and Numerical Evidence for Improving Confusing Legal Judgment Prediction. *CoRR* abs/2211.08238 (2022). https://doi.org/10.48550/arXiv.2211.08238 arXiv:2211.08238
[16] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An Embedding Learning Framework for Numerical Features in CTR Prediction. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 2910–2918. https://doi.org/10.1145/3447548.3467077
[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 1725–1731. https://doi.org/10.24963/ijcai.2017/239
[18] Shagufta Henna and Shyam Krishnan Kalliadan. 2021. Enterprise Analytics using Graph Database and Graph-based Deep Learning. *CoRR* abs/2108.02867 (2021). arXiv:2108.02867 https://arxiv.org/abs/2108.02867
[19] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 585–593. https://doi.org/10.1145/3534678.3539381
[20] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 2333–2338. https://doi.org/10.1145/2505515.2505665
[21] Zhaoheng Huang, Zhicheng Dou, Yutao Zhu, and Zhengyi Ma. 2022. MCP: Self-supervised Pre-training for Personalized Chatbots with Multi-level Contrastive Sampling. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 1030–1042. https://doi.org/10.18653/v1/2022.findings-emnlp.73
[22] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 39–48. https://doi.org/10.1145/3397271.3401075
[23] Weize Kong, Swaraj Khadanga, Cheng Li, Shaleen Kumar Gupta, Mingyang Zhang, Wensong Xu, and Michael Bendersky. 2022. Multi-Aspect Dense Retrieval. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 3178–3186. https://doi.org/10.1145/3534678.3539137
[24] Minghan Li and Jimmy Lin. 2021. Encoder Adaptation of Dense Passage Retrieval for Open-Domain Question Answering. *CoRR* abs/2110.01599 (2021). arXiv:2110.01599 https://arxiv.org/abs/2110.01599
[25] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.* 14, 1 (2020), 50–60. https://doi.org/10.14778/3421424.3421431
[26] Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. TwinBERT: Distilling Knowledge to Twin-Structured BERT Models for Efficient Retrieval. *CoRR* abs/2002.06275 (2020). arXiv:2002.06275 https://arxiv.org/abs/2002.06275
[27] Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 1211–1225. https://aclanthology.org/2023.findings-emnlp.86
[28] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 176–186. https://doi.org/10.1145/3477495.3531961
[29] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (Eds.). ACM, 19–34. https://doi.org/10.1145/3183713.3196926
[30] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE ACM Trans. Audio Speech Lang. Process.* 24, 4 (2016), 694–707. https://doi.org/10.1109/TASLP.2016.2520371
[31] Charuta Pande, Hans Friedrich Witschel, and Andreas Martin. 2021. New Hybrid Techniques for Business Recommender Systems. *CoRR* abs/2109.13922 (2021). arXiv:2109.13922 https://arxiv.org/abs/2109.13922
[32] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, Jian Su, Xavier Carreras, and Kevin Duh (Eds.). The Association for Computational Linguistics, 2249–2255. https://doi.org/10.18653/v1/d16-1244
[33] Zhen Qin, Zhongliang Li, Michael Bendersky, and Donald Metzler. 2020. Matching Cross Network for Learning to Rank in Personal Search. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 2835–2841. https://doi.org/10.1145/3366423.3380046
[34] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410
[35] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu (Eds.). IEEE Computer Society, 995–1000. https://doi.org/10.1109/ICDM.2010.127
[36] Qusai Shambour and Jie Lu. 2015. An effective recommender system by unifying user and item trust information for B2B applications. *J. Comput. Syst. Sci.* 81, 7 (2015), 1110–1126. https://doi.org/10.1016/j.jcss.2014.12.029
[37] Hongyu Shan, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Chenliang Li. 2023. Beyond Two-Tower: Attribute Guided Representation Learning for Candidate Retrieval. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 3173–3181. https://doi.org/10.1145/3543507.3583254
[38] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1441–1450. https://doi.org/10.1145/3357384.3357895
[39] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Attentive Multi-View Learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 3863–3869. https://doi.org/10.24963/ijcai.2019/536
[40] Dianshuang Wu, Guangquan Zhang, and Jie Lu. 2015. A Fuzzy Preference Tree-Based Recommender System for Personalized Business-to-Business E-Services. *IEEE Trans. Fuzzy Syst.* 23, 1 (2015), 29–43. https://doi.org/10.1109/TFUZZ.2014.2315655
[41] Jingyuan Yang, Chuanren Liu, Mingfei Teng, Ji Chen, and Hui Xiong. 2018. A Unified View of Social and Temporal Modeling for B2B Marketing Campaign Recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 5 (2018), 810–823. https://doi.org/10.1109/TKDE.2017.2783926
[42] Jingyuan Yang, Chuanren Liu, Mingfei Teng, Hui Xiong, March Liao, and Vivian Zhu. 2015. Exploiting Temporal and Social Factors for B2B Marketing Campaign Recommendations. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, Charu C. Aggarwal, Zhi-Hua Zhou, Alexander Tuzhilin, Hui Xiong, and Xindong Wu (Eds.). IEEE Computer Society, 499–508. https://doi.org/10.1109/ICDM.2015.71
[43] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. 2021. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information*

Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 28798–28810. https://proceedings.neurips.cc/paper/2021/hash/f18a6d1cde4b205199de8\729a6637b42-Abstract.html

[44] Dezhong Yao, Yuhong Gu, Gao Cong, Hai Jin, and Xinqiao Lv. 2022. Entity Resolution with Hierarchical Graph Attention Networks. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 429–442. https://doi.org/10.1145/3514221.3517872

[45] Jing Yao, Zheng Liu, Junhan Yang, Zhicheng Dou, Xing Xie, and Ji-Rong Wen. 2023. CDSM: Cascaded Deep Semantic Matching on Textual Graphs Leveraging Ad-hoc Neighbor Selection. *ACM Trans. Intell. Syst. Technol.* 14, 2 (2023), 32:1–32:24. https://doi.org/10.1145/3573204

[46] Wenpeng Yin and Hinrich Schütze. 2015. MultiGranCNN: An Architecture for General Matching of Text Chunks on Multiple Levels of Granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 63–73. https://doi.org/10.3115/v1/p15-1007

[47] Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. 2018. Neural Ranking Models with Multiple Document Fields. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 700–708. https://doi.org/10.1145/3159652.3159730

[48] Qi Zhang, Qinglin Jia, Chuyuan Wang, Jingjie Li, Zhaowei Wang, and Xiuqiang He. 2021. AMM: Attentive Multi-field Matching for News Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1588–1592. https://doi.org/10.1145/3404835.3463232

[49] Xuirui Zhang and Hengshan Wang. 2005. Study on recommender systems for business-to-business electronic commerce. *Communications of the IIMA* 5, 4 (2005), 8.

[50] Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. 2021. TextGNN: Improving Text Encoder via Graph Neural Network in Sponsored Search. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 2848–2857. https://doi.org/10.1145/3442381.3449842

[51] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 2780–2791. https://doi.org/10.1145/3459637.3482243

[52] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2023. Large Language Models for Information Retrieval: A Survey. *CoRR* abs/2308.07107 (2023). arXiv:2306.07401 https://arxiv.org/abs/2308.07107