Matching of Descriptive Labels to Glossary Descriptions

Anonymous ACL submission

Abstract

Semantic text similarity plays an important role in software engineering tasks in which engineers are requested to clarify the semantics of descriptive labels (e.g., business terms, table column names) that are often consists of too short or too generic words and appears in their IT systems. We formulate this type of problem as a task of matching descriptive labels to glossary descriptions. We then propose a framework to leverage an existing semantic text similarity measurement (STS) and augment it using semantic label enrichment and set-based collective contextualization where the former is a method to retrieve sentences relevant to a given label and the latter is a method to compute similarity between two contexts each of which is derived from a set of texts (e.g., column names in the same table). We performed an experiment on two datasets derived from publicly available data sources. The result indicated that the proposed methods helped the underlying STS correctly match more descriptive labels with the descriptions.

1 Introduction

In general IT projects, such as database and business process migration, IT engineers invest significant effort in verifying consistency between various models, such as table schemata and diagrams that depict object relations. Maintaining a glossary of domain terms is a best practice that helps alleviate their workload. The glossary serves as a reference that maps descriptive labels (such as business terms and table column names) to corresponding glossary descriptions, which are typically English sentences that provide explanations and contexts.

Target Problem and Challenges

In this paper, we address such a common mapping problem between descriptive labels and glossary descriptions. We refer to this problem as Descriptive Labels to Descriptions (DLD). In DLD, we are given multiple datasets and glossaries. Each dataset contains a list of descriptive labels, while each glossary contains a list of glossary descriptions. The goal is to establish mappings from each label to its corresponding description.

However, there are several technical challenges in mapping descriptive labels to the glossary descriptions due to the nature of the descriptive labels such as too short and too generic words included by the descriptive names.

Our Approach and Research Questions

To tackle the technical challenges, in this paper, we propose a novel framework to solve DLD problems effectively by enriching the short and/or generic words and capturing the context of the generic and/or ambiguous words. Our framework is designed to be flexible enough to employ variations of underlying semantic text similarity (STS) models, enrichment methods, and contextualization methods whereas our implementation is limited to reasonable combinations of a traditional TFIDF model, the PromCSE model (Jiang et al., 2022) (BERT-based STS model), the Flan-T5 (Chung et al., 2022) large language model (LLM), and Wikidata. With our implementation we performed an experiment on two glossaries: a business glossary derived from the financial industry business ontology and more than 1000 pairs of column names and corresponding descriptions obtained from the Kaggle webpages. We organized our experiment to verify our hypothesis that there are many descriptive labels that cannot be mapped to corresponding descriptions by commonly used text similarity models due to cryptic words and our label enrichment and contextualization methods help the underlying STS models work better for such problematic labels. More specially, the research questions we address in this paper are as follows.

- How much improvement is observed in the

metrics Mean Reciprocal Rank (MRR) and Hits@k with the label enrichment methods?

- How much improvement is observed in MRR and Hits@k with the label contextualization methods?
- What kind of labels meet the out-of-vocabulary issue and the ambiguity issue? And what kind of labels can be solved by the label enrichment methods, and can be disambiguated by label contextualization methods?

The reason why we use the Mean Reciprocal Rank (MRR) and Hits@k is that the DLD problem can be seen as a task of ranking descriptions corresponding to given descriptive labels and these metrics are commonly used for recommend systems.

Contributions

Our contributions in this paper are as follows.

- We formulate the DLD problem, and identified the technical issues: out-of-vocabulary issue and ambiguity issue of cryptic words. We provide two practical benchmark datasets: Kaggle and FIBO including these issues.
- To solve out-of-vocabulary issue, we propose Label Semantic Enrichment method which leverages external knowledge.
- To solve ambiguity issue, we propose Set-based Collective Contextualization method by leveraging Large Language Model.
- In our experiments, we clarified how effectively our approaches solve the issues.

The rest of this paper is organized as follows. In the next two sections, we discuss related works, and provide motivating examples using practical datasets. We then present our approach to address them effectively, and formulate the DLD task. Subsequently, we demonstrate how the DLD task is tackled using several approaches, presenting experimental results. Afterward, we provide a comprehensive analysis of our experimental results. Finally, we conclude the paper.

2 Related Work

In general, DLD is a task of linking a chunk of text in a group to one in another group. In this sense, it could be considered as a natural language processing problem, specifically semantic text similarity or entity linking (Shen et al., 2014, 2021). Also, we could consider it as a problem of aligning between domain-specific labels or descriptive names, as a set of domain concepts, to another ontology of a common glossary.

Semantic Text Similarity

There have been many studies on STS (Chandrasekaran and Mago, 2021). Some exploited general knowledge bases (Li et al., 2020), as in our study. Such knowledge-based methods measure the similarity of two terms on the basis of the structural properties of the knowledge bases, such as the number of edges. Our method, however, uses the knowledge bases only to enrich descriptive names.

Other methods are corpus based such as word2vec (Mikolov et al., 2013) and BERT (Devlin et al., 2019). Such methods leverage large corpora to compute word-embedding useful for measuring the similarity between terms on the basis of the idea that similar words occur together. The same idea is also applied for capturing the characteristics of sentences, as with Sentence-BERT (Reimers and Gurevych, 2019) and PromCSE (Jiang et al., 2022). Many search engines use this type of method to retrieve and rank relevant sentences and webpages. We designed our method in such a way that it benefits from the advances of the corpus-based methods and pre-trained models.

Entity Linking and Ontology Learning

Entity linking (Shen et al., 2014), such as ColNet (Chen et al., 2019) and TabEL (Bhagavatula et al., 2015), is a task of linking terms described in a document to entities defined in a knowledge graph such as Wikidata. Ontology learning (Asim et al., 2018), matching (Shvaiko and Euzenat, 2011), or alignment (Ardjani et al., 2015) are similar tasks that automatically or semi-automatically gather terms and relations from documents to create an ontology and discover correspondence between ontologies. Background knowledge is known to significantly improve the performance of ontology matching systems (Portisch et al., 2021). Compared with these problems, the DLD problem mentions neither knowledge graph nor ontologies. The descriptive names in a DLD problem are not within contextual sentences or linked to other concepts, entities, or values. Glossaries as ontologies in a DLD problem are also not linked but just descriptions of known concepts.

Named Entity Disambiguation

Named entity disambiguation on knowledge graphs including ontologies is a key component for the success of semantic text similarity, entity linking, and ontology learning. Likewise, our label enrichment method and set-based collective contextualization are both considered as methods to disambiguate the descriptive labels for solving the DLD problems. As in the case of our label enrichment method, recent literature (Mulang' et al., 2020; Bastos et al., 2021) also leverage the triples obtained from Wikidata to improve the performance of pre-trained models for the named entity disambiguation on Wikipedia whereas we leverage the sentences obtained from Wikidata. In addition, unlike our contextualization, they do not compute the context of a set of entities and rely only on the context of each single entity.

The idea of using a knowledge base for disambiguation is also presented in the literature (Ho et al., 2010). It proposes the use of a knowledge base to measure the similarity between texts for semantic text similarity and integrate it with a corpus-based measurement. However, it does not include any process for enriching and/or contextualizing given texts.

3 Motivating Examples

In software engineering tasks such as database and business process migrations, engineers are often requested to clarify the semantics of descriptive labels (e.g., business terms, table column names). For example, during the migration from a legacy system, IT engineers thoroughly analyze the original descriptive labels present in the existing data models. They then create a new logical data model and establish mappings between the original labels and the new ones. Understanding the semantics of the original labels in the context of the new logical data model is crucial, and the glossary serves as a valuable resource. However, the original data model often suffers from incompleteness and inconsistencies, as its element names may differ from those in the new logical data model, and an up-to-date or accessible dictionary may be lacking. Moreover, the mapping process typically requires human involvement and may not have an initial mapping in the first iteration.

We could leverage a state-of-the-art semantic text similarity (STS) measurement to map each descriptive labels to a corresponding description. However, the following nature of the descriptive names makes it difficult to make the mapping using the STS measurement in high accuracy.

- The descriptive labels (e.g., LOAN_AMT, ACS_DT) often contain cryptic (too short, too generic, and too ambiguous) words such as AMT (amount), ACS (access), and DT (date). These cryptic words are out-of vocabulary unlike aliases or nicknames, and often makes it difficult to understand their meanings.

- The descriptive labels often appear only in database table schemata or program variables. This prevents us from associating the descriptive labels with documents.

- Many descriptive labels used in the IT systems are specific to those IT systems. We could not rely on mappings between the descriptive labels and descriptions created for other IT systems.

The same situation arises even in standardized or commonly used domain-specific ontologies such as financial industry business ontology (FIBO). FIBO is an ontology for financial business applications. It defines a named entity "ALL" the description of which is "the currency identifier for Lek (the currency of Albania)". However, it is a very general word and often used as in the case of "all types of bank loans". Likewise, we found many descriptions that include the word "all". Therefore, no STS model is effective for matching the descriptive name with the correct definition.

As a preliminary experiment, we collected the named entities and corresponding descriptions from FIBO, and ran PromCSE to compute the similarity scores between all the pairs of the named entities and descriptions. The top-ranked descriptions corresponding to "ALL" were

- "collection representing the total membership, or úniverse, of people, resources, products, services, events, or entities of interest for some question, experiment, survey or statistical program" (0.309),
- "location in physical space" (0.291),
- "a collection of managed investments that are all managed by a single investment institution" (0.274),

where the values enclosed with the parentheses are the similarity scores reported from PromCSE. The

correct description of "ALL" was ranked 490th (0.063).

4 Approach

Our approach is as follows:

- We employ a STS model to measure a similarity of two sentences.
- We apply Label Semantics Enrichment module to enrich descriptive labels.
- We take into account the context of both of descriptive labels and glossary descriptions.

Label Semantics Enrichment (LSE)

A basic strategy to measure a similarity between a descriptive label and a glossary description is using STS models such as TFIDF, PromCSE and LLM. TFIDF suffer from aliases because it's built on word-level exact matching. PromCSE and LLM also may suffer from minor aliases and cryptic words in descriptive labels.

Furthermore we employ Label Semantics Enrichment (LSE) to solve this problem. LSE module retrieves sentences relevant to the given the descriptive label by using a external knowledge database such as Wikidata and Bing. If the retrieved sentences include sufficiently various aliases and relevant phrases of cryptic words, STS model will work more effectively.

Set-based Collective Contextualization (SCC)

As mentioned earlier, descriptive labels can often be ambiguous. In the case that two tables have columns with the same label but semantically different meanings, the same glossary description will be assigned to columns that are semantically different.

To address this problem, we propose incorporating the context of a set of descriptive labels and a set of glossary descriptions. For example, when matching column names from multiple tables to various glossaries, we consider the collective context provided by a set of column names within the same table and a set of glossary descriptions within the same glossary. By leveraging this broader context, we can more effectively identify and disambiguate the intended meanings of the columns.

5 Problem Setting

Now we are given a dataset $\mathcal{D}$ which contains several semantic groups $D_i \in \mathcal{D}$. The i-th group $D_i = (L_i, G_i)$ has a set of descriptive labels $L_i = \{l_{i,1}, l_{i,2}, \cdots, l_{i,n}\}$ and a glossary (a set of glossary descriptions) $G_i = \{g_{i,1}, g_{i,2}, \cdots, g_{i,n}\}$. $l_{i,p} \in L_i$ is p-th descriptive label of L_i. $g_{i,p} \in G_i$ is p-th glossary description of G_i. $g_{i,p}$ describes a meaning of $l_{i,p}$.

The ultimate goal is to establish a complete mapping between $(l_{i,p}, L_i)$ and its corresponding $(g_{i,p}, G_i)$. However, the difficulty of this problem is heavily influenced by the number of groups D_i and the size of the label set L and the glossary G in each group D.

To standardize the difficulty, we consider N-choice problem here. In this scenario, given a target label set L and a target label $l \in L$, the objective is to identify the corresponding pair of the target description and its glossary (g, G) from N candidates $\{(g_k, G_k)\}_{k=1}^{N}$. This is done by measuring the similarity between the label side (l, L) and the glossary side (g_k, G_k).

6 DLD Framework

In our framework, DLD-Similarity Score Ψ between label side (l, L) and glossary side (g, G) is defined as

$$\begin{aligned}\Psi(l, g, L, G|\theta) &= \Psi_{\mathrm{T}}(l, g, L|\theta_{\mathrm{LSE}}, \theta_{\mathrm{STS}}) \\ &\times \Psi_{\mathrm{C}}(L, G|\theta_{\mathrm{SCC}}).\end{aligned}$$

Here Ψ_{T} is Text-Similarity Score, and Ψ_{C} is Context-Similarity Score. In Ψ_{T}, $\theta_{\mathrm{LSE}} \in \{\mathrm{on}, \mathrm{off}\}$ is a switch to enable or disable LSE module, and $\theta_{\mathrm{STS}} \in \{\mathsf{T}, \mathsf{P}, \mathsf{L}\}$ is a switch indicating which STS model is used from three variations: TFIDF, PromCSE, and LLM. In Ψ_{C}, $\theta_{\mathrm{SCC}} \in \{\mathrm{on}, \mathrm{off}\}$ is a switch to enable or disable SCC module.

Text-Similarity Score

Text-Similarity Score Ψ_{T} measures a similarity between descriptive label l and glossary description g. It's defined as

$$\Psi_{\mathrm{T}}(l, g, L|\theta_{\mathrm{LSE}}, \theta_{\mathrm{STS}}) = \max_{s \in \mathrm{LSE}(l|\theta_{\mathrm{LSE}})} \mathrm{STS}(s, g, L|\theta_{\mathrm{STS}}).$$

Ψ_{T} receives a set of descriptive labels L, a target descriptive label $l \in L$, and a glossary description g as inputs. It collects sentences $\{s\}$ relevant to l from the external knowledge database by invoking $\mathrm{LSE}(l)$, and computes similarity score between the each collected sentence s and g by invoking STS, and outputs max of them as Text-Similarity Score. L is used as contextual information in STS.

We have two variations of LSE function: enabled version and disabled version. The enabled version $\mathrm{LSE}(l|\mathrm{on})$ collects sentences $\{s_i\}$ relevant to l from external knowledge, and returns them. The disabled version $\mathrm{LSE}(l|\mathrm{off})$ just returns l.

We also have three variations of STS function: TFIDF version $\mathrm{STS}(\cdot|\mathsf{T})$, PromCSE version $\mathrm{STS}(\cdot|\mathsf{P})$, and LLM version $\mathrm{STS}(\cdot|\mathsf{L})$. See 7 for more details.

Context-Similarity Score

Context-Similarity Score Ψ_{C} measures a contextual similarity between L and G. We have two variations: SCC enabled version and disabled version.

The SCC enabled version $\Psi_{\mathrm{C}}(L, G|\mathrm{on})$ directly ask to LLM about a probability of L and G being the same or different. See 7 for more details.

The disable version $\Psi_{\mathrm{C}}(L, G|\mathrm{off})$ is always outputs 1.

7 Implementation Details

We implemented our method using Python. It leverages Wikidata as external knowledge in LSE module. It runs TFIDF, PromCSE and Flan-T55-xxl (11B) in STS module to measure a similarity score between two sentences, and also uses Flan-T5-xxl (11B) in SCC module to measure Context-Similarity Score between a set of labels and a set of descriptions. In our preliminary analysis, we investigated performance of 4 LLMs (Flan-T5, GPT2, Bloom-Z and LLama2), and Flan-T5 was the best for our task. The use of TFIDF and PromCSE is straightforward, therefore, in the following sections, we describe about how to use Wikidata in LSE module, and how to use Flag-T5 in STS and SCC module.

Using Wikidata for LSE

In LSE module, we used Wikidata as external resource. The webpage of Wikidata provides a search interface that we usually access using a Web browser. We leverage this Web interface for the implementation. It sends queries of descriptive labels to the search interface using the HTTP protocol, and parses resulting webpages to extract entity IDs.

Wikidata also provides a SPARQL (Harris and Seaborne, 2013) endpoint as a query service. We leverage this query service to collect label names and descriptions of the collected entity IDs. We also use the `label` property[1] and `description` property[2] to generate sentences relevant to the query phrase.

LLM-based Semantic Text Similarity Model

We employed three STS models (TFIDF, PromCSE, and LLM) to compute the Text-Similarity Score Ψ_{T}. TFIDF is well-known technique. PromCSE is the state-of-the-art embedding-based technique, and it was better than SBERT in our preliminary analysis. TFIDF and PromCSE can directly generate similarity scores for the given pair of sentences. LLM requires a natural language prompt as input, and also generates a natural language answer to be translated into numerical score.

We performed prompt engineering to create suitable inputs for LLM and developed a method to extract scores from the generated answer. Figure 1 shows the typical prompt example.

To translate the LLN answer to numerical score, $\mathrm{STS}(\cdot|\mathsf{L})$ function collects top N tokens $\mathscr{T} = \{t_i\}_{i=1}^{N}$ with their probability $p(t_i)$ from LLM answer. It classifies the N tokens into a set of "yes" tokens $\mathscr{T}_{\mathrm{y}}$, "no" tokens $\mathscr{T}_{\mathrm{n}}$, and others. And computes probability ratio of "yes" and "no" as $s = p_{\mathrm{y}}/(p_{\mathrm{y}} + p_{\mathrm{n}})$. Here p_{y} and p_{n} is sum of probability which token is "yes" and "no". They can be computed as follows:

$$p_{\mathrm{y}} = \sum_{t \in \mathscr{T}_{\mathrm{y}}} p(t).$$

LLM-based Context-Similarity Algorithm

We also employed LLM to measure the Context-Similarity between a set of labels and a set of descriptions. Figure 2 shows the typical prompt example. A scoring logic is same to the logic mentioned in 7.

8 Experimental Setup

As we described in Section 1, our experiment is organized to verify the hypothesis that (1) there are many descriptive labels that cannot be mapped to corresponding descriptions by commonly used text similarity models due to cryptic words and (2) our label enrichment and contextualization methods help the underlying STS models work better for such problematic labels. In this section, we first describe how we prepared the datasets based

[1] http://www.w3.org/2000/01/rdf-schema#label
[2] https://schema.org/description

```
I have a dataset and a glossary. The given dataset has these columns.
[Column names]
  - School District Code
  - County Code
  ...
[Question]
Is "School District Code" same to the following concept in glossary?
glossary description: "The code by which a school district is identified, as utilized by the Department's ..."
[Answer (Yes/No)]
```

Figure 1: Prompt example for LLM-based STS

```
I have a dataset and a glossary. The given dataset has these columns.
[Column names]
  - ordered
  - device_computer
  ...
The given glossary has these glossary terms.
[Glossary terms]
  - Indicates whether an appeal to the published decision has been received.
  - Indicates if Design Review is part of the application process for this permit.
  ...
[Question]
Does these glossary terms describe the given column names?
[Answer (Yes/No)]
```

Figure 2: Prompt example for SCC

on the publicly available data sources. We then describe what metrics and why we used and how we compared the different combinations of the STS models, the label enrichment method, and the contextualization method.

Dataset and Benchmark

Our experiment is performed on the two datasets derived from the Kaggle webpages and the financial industry business ontology (FIBO). Table 1 shows the statistics of these datasets.

The Kaggle dataset consists of 85 semantic groups that includes 1347 descriptive labels and corresponding descriptions in total where we consider tables and column names we extracted from the Kaggle webpages as the semantic groups and the descriptive labels. The smallest semantic group contains only 10 descriptive labels while the largest semantic group contains 36 descriptive labels. We further investigated how many semantic groups include each descriptive label to see if the dataset is suitable to evaluate the effectiveness of the contextualization method. Table 2 summarizes the descriptive labels and the numbers of corresponding semantic groups, which are referred to as frequencies of the descriptive labels, where we selected only the descriptive labels whose frequency is more than 7. For example, the descriptive label "type" appears in 7 semantic groups and has different meanings such as "type of wine", "media type of animation film" and "flag if company is private or public".

FIBO defines concepts and relations used in financial domain, using the Web Ontology Language (OWL) (Smith et al., 2004). It consists of 2086 named entities each of which has its label and description specified by particular XML tags such as the `description` and `definition`. We collected these labels and descriptions as descriptive labels and descriptions of the dataset. We then partitioned the pairs of the descriptive labels and descriptions into 44 semantic groups based on IRIs of the corresponding named entities. Unlike the Kaggle dataset, there is no descriptive label that is included by multiple semantic groups.

From these two datasets, we created 4 problems: Kaggle-10-choice, Kaggle-50-choice, FIBO-10-choice, FIBO-50-choice as mentioned in Section 5 and performed the experiment on these 4 problems.

Table 1: Statistics of the datasets

	Kaggle	FIBO
entries	1347	2086
semantic groups	85	44
avg of # of words (label)	1.64	4.05
max of # of words (label)	6	16
min of # of words (label)	1	1
avg of # of words (desc)	14.4	16.1
max of # of words (desc)	389	140
min of # of words (desc)	2	3

Evaluation Method

In our evaluation, we compare the 12 combinations of models based on the 3 underlying STS models (TFIDF, PromCSE, LLM-based) and the presence or absence of the label semantics enrichment

Table 2: Frequency of descriptive labels of Kaggle

label	frequency	label	frequency
name	16	country	8
date	14	title	8
age	12	id	8
year	9	type	7
county	9	status	7
gender	8	description	7

(SLE) and the set-based collective contextualization (SCC), which are represented by the following naming rule: $\{\mathtt{T}, \mathtt{P}, \mathtt{L}\}$-$\{\phi, \mathtt{LSE}\}$-$\{\phi, \mathtt{SCC}\}$, where T,P, and L represent TFIDF, PromCSE, and the LLM-based STS. For example, 'T-LSE-SCC' represents the combination of TFIDF, LSE, and SCC. 'L' represents the LLM-based STS model not augmented with LSE or SCC.

To measure the success of matching the descriptive labels with the descriptions, we employ the Mean Reciprocal Rank (MRR) and Hits@k as performance metrics which are commonly used for evaluating the performance of search engines and recommendation systems. This is because the DLD problem can be seen as a task of ranking descriptions corresponding to given descriptive labels.

9 Experimental Results

We present our comprehensive results against the 10-choice problems of the Kaggle and FIBO datasets in Table 3 and against the 50-choice problems in Table 4.

As a whole, regarding the first and second research questions, we proved that the label semantics enrichment (LSE) and the set-based collective contextualization (SCC) helps the underlying STS models produce better scores in both MRR and Hits@k except that LSE often gave negative impact to PromCSE and the LLM-based STS model in many cases.

Regarding the third research question, we observed that there were reasonably many labels not correctly mapped only by the underlying STS models were correctly mapped by the STS model augmented with LSE and SCC. In particular, SCC successfully disambiguated generic words, and helped the underlying STS model correctly mapped out-of-vocabulary labels which include highly cryptic words. In addition, the L-SCC combination achieved over 99% in MRR for the 10-choice problem. The failed descriptive labels were difficult to be correctly mapped even by humans.

We describe more details in the following sections one by one.

Table 3: Result of 10-choice problem

Kaggle10C	MRR	Hits@1	Hits@3	Hits@5
T	0.735	0.670	0.687	0.687
T-LSE	0.795	0.733	0.788	0.818
T-SCC	0.741	0.682	0.687	0.687
T-LSE-SCC	0.864	0.827	0.854	0.860
P	0.872	0.800	0.931	0.972
P-LSE	0.865	0.794	0.918	0.963
P-SCC	0.987	0.978	0.996	0.999
P-LSE-SCC	0.986	0.977	0.996	0.997
L	0.982	0.970	0.995	0.997
L-LSE	0.975	0.958	0.993	0.997
L-SCC	0.994	**0.990**	**0.998**	**0.999**
L-LSE-SCC	**0.994**	**0.990**	**0.998**	**0.999**
FIBO10C	MRR	Hits@1	Hits@3	Hits@5
T	0.751	0.689	0.716	0.727
T-LSE	0.874	0.833	0.881	0.907
T-SCC	0.768	0.712	0.729	0.733
T-LSE-SCC	0.925	0.898	0.933	0.937
P	0.941	0.911	0.963	0.986
P-LSE	0.957	0.934	0.973	0.990
P-SCC	0.988	0.978	0.999	**1.000**
P-LSE-SCC	0.990	0.981	**1.000**	**1.000**
L	0.988	0.978	0.998	**1.000**
L-SCC	**0.993**	**0.986**	**1.000**	**1.000**

This table shows the experiment results under the 12 settings (STS (TFIDF, PromCSE, LLM), LSE on/off, SCC on/off) with Kaggle and FIBO datasets. Hit@k is hit ratio represents how many times the true descriptions appeared in top-k in the list. Larger is better. MRR is mean reciprocal rank. Larger is better. Highest scores of each metric are denoted in **bold**.

Effectiveness of LSE

We observed that LSE was effective only for TFIDF models as shown in Table 5, but the negative impact to the other models are very limited. We think that the pre-trained models, PromCSE and LLM, already have better or equivalent capability compared with LSE.

Table 6 shows some examples of descriptive labels whose rankings were improved or degraded by LSE from the results of Kaggle-10-choice problem. It shows that LSE properly address the out-of-vocabulary problem encountered in the TFIDF

Table 4: Result of 50-choice problem

Kaggle50C	MRR	Hits@1	Hits@5	Hits@10
T	0.654	0.604	0.686	0.687
T-SLE	0.698	0.635	0.753	0.776
T-SCC	0.691	0.671	0.687	0.687
T-SLE-SCC	0.816	0.782	0.842	0.849
P	0.726	0.624	0.846	0.916
P-SLE	0.728	0.638	0.836	0.906
P-SCC	0.960	0.933	0.990	0.995
P-SLE-SCC	0.957	0.928	0.989	0.993
L	0.939	0.903	0.984	0.993
L-SCC	**0.982**	**0.970**	**0.993**	**0.996**
FIBO50C	MRR	Hits@1	Hits@5	Hits@10
T	0.683	0.648	0.701	0.712
T-SLE	0.810	0.772	0.848	0.868
T-SCC	0.707	0.677	0.720	0.728
T-SLE-SCC	0.877	0.841	0.916	0.928
P	0.870	0.824	0.923	0.952
P-SLE	0.907	0.871	0.950	0.969
P-SCC	0.953	0.918	0.995	0.998
P-SLE-SCC	0.965	0.938	0.998	**1.000**
L	0.958	0.932	0.992	0.996
L-SCC	**0.976**	**0.957**	**0.999**	**1.000**

model, thereby successfully enriched cryptic or domain specific words. Additionally, we provide some examples of descriptive labels from the Kaggle result that were accurately ranked as top 1 by the LSE-enhanced model (T-LSE), but failed to achieve the same ranking by the baseline model (T) in Table 6.

On the other hand, there were certain descriptive labels that were successfully ranked by the baseline model (T), but failed by the LSE-enhanced model (T-LSE). Table 6 shows some examples. For more general words, the incorporation of LSE may introduce noise and hinder the accurate identification of the corresponding glossary description.

Table 5: Improved MRR by LSE in Kaggle10

w/o LSE	MRR	w/ LSE	MRR	improved
T	0.735	T-LSE	0.795	0.061
T-SCC	0.741	T-LSE-SCC	0.864	**0.123**
P	0.872	P-LSE	0.865	-0.007
P-SCC	0.987	P-LSE-SCC	0.986	-0.001
L	0.982	L-LSE	0.975	-0.007
L-SCC	0.994	L-LSE-SCC	0.994	0.000

Table 6: Example descriptive labels effected by LSE

Improved labels by LSE
'AADT', 'Burglary', 'chlorides', 'DEROG', 'FGA', 'FTA', 'HAZMAT', 'isbn', 'isFlaggedFraud', 'iso3', 'lvdd', 'MentHlth', 'MVP', 'Riot', 'rpm', 'Sugarcanes', 'synopsis', 'total sulfur dioxide', 'Ward Interactions'

Degraded labels by LSE
'Age', 'category', 'Category', 'country', 'date', 'description', 'Genre', 'id', 'Make', 'name', 'Rating', 'Score', 'sex', 'status', 'Status', 'Title', 'type', 'url'

Effectiveness of SCC

Table 7 shows the improvement of the MRR scores by SCC where SCC improved the MRR score in all the cases. In particular, it contributed to the improvement of over 10% for the PromCSE models.

We analyzed the descriptive labels that were successfully ranked as top 1 as the result of the improvement by SCC (P-LSE-SCC), but failed by the baseline model (P-LSE). We found these descriptive labels can be classified into two groups: general words and highly cryptic words. For general words, PromCSE collected several descriptions from different glossaries with high confidences. SCC worked as screening indicators in this situation. Table 8 shows some examples of the general labels whose rankings were improved by SCC.

In the case of highly cryptic words, PromCSE often struggled to find similar descriptions from all candidates, because the highly cryptic words are out-of-vocabulary even for PromCSE. Consequently, the Text-Similarity Scores of the top-ranked descriptions were relatively low. However, the Context-Similarity Score of the corresponding glossary was significantly high. SCC elevated the confidence of these correct but low-ranked descriptions, pushing them to the top of the list. Table 8 also shows some examples of the cryptic labels improved by SCC.

Table 7: Improved MRR by SCC in Kaggle10

w/o SCC	MRR	w/ SCC	MRR	improved
T	0.735	T-SCC	0.741	0.006
T-LSE	0.795	T-LSE-SCC	0.864	0.069
P	0.872	P-SCC	0.987	0.115
P-LSE	0.865	P-LSE-SCC	0.986	**0.122**
L	0.982	L-SCC	0.994	0.011
L-LSE	0.975	L-LSE-SCC	0.994	0.019

Table 8: Example descriptive labels improved by SCC

General labels
'Age', 'close', 'Close', 'date', 'Gender', 'high', 'id', 'ID', 'name', 'Name', 'Pos', 'Services', 'Status', 'Tags', 'Title', 'TITLE', 'type', 'Type', 'url', 'Value', 'y', 'Year', 'Years'

Cryptic labels
'3PA', 'ACS/Map', 'AST', 'CholCheck', 'CLAGE', 'CLNO', 'DEROG', 'DREB', 'Fedu', 'FGS', 'FTA', 'FTM', 'G1', 'G2', 'G3', 'GD', 'GF', 'GO / SC Num', 'hsc_p', 'MentHlth', 'Mjob', 'OREB', 'PTS', 'RC_ID', 'RH', 'SFY'

10 Concluding Remarks

We formulated the DLD problem as important and practical task, and identified the technical issues: out-of-vocabulary issue and ambiguity issue of cryptic words. We proposed a framework to solve the issues. To solve out-of-vocabulary issue, we proposed Label Semantic Enrichment method by leveraging external knowledge. To solve ambiguity issue, we proposed Set-based Collective Contextualization method by leveraging Large Language Model. We provided two practical benchmark datasets: Kaggle and FIBO including the issues, and designed N-choice problem on the datasets. In our experiments, we clarified how our approach are effective to solve the issues. We plan to release the benchmark datasets under a reasonable license for future advances in this technical area.

Limitations

The empirical evaluation of our methods is mainly done on the datasets derived from the publicly available data sources whereas we used pre-trained models of Flan-T5 and PromCSE in the evaluation. Therefore, there might be overlapping data sources, and hence the risk of data leakage. Even so, our evaluation showed that both the label enrichment and the contextualization contributed to the improvement of the TFIDF-based STS, which never rely on any external data sources.

In our experiment, the LLM-based STS model outperformed the other models in all the cases. However, we need to care about its inference time when we use it in a practical situation, since the estimated total inference time for completing the experiment was roughly 150 hours. On the other hand, the TFIDF and PromCSE models were obviously more efficient than the LLM-based STS model. Those total inference times based on our observation were 1.6 hours and 25 hours, respectively.

References

Fatima Ardjani, Djelloul Bouchiha, and Mimoun Malki. 2015. Ontology-alignment techniques: Survey and analysis. *International Journal of Modern Education and Computer Science*, 7:67–78.

Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database*, 2018.

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang, Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. Recon: relation extraction using knowledge graph context in a graph neural network. In *Proceedings of the Web Conference 2021*, pages 1673–1685.

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: Entity linking in web tables. In *International Semantic Web Conference*, pages 425–441. Springer.

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity – a survey. *ACM Computing Surveys*, 54(2).

Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. 2019. ColNet: Embedding the semantics of web tables for column type prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 29–36.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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*, pages 4171–4186. Association for Computational Linguistics.

Steve Harris and Andy Seaborne. 2013. Sparql 1.1 query language. https://www.w3.org/TR/sparql11-query/.

Chukfong Ho, Masrah Azrifah Azmi Murad, Rabiah Abdul Kadir, and Shyamala C Doraisamy. 2010. Word sense disambiguation-based sentence similarity. In *Coling 2010: Posters*, pages 418–426.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning.

Fei Li, Lejian Liao, Lanfang Zhang, Xinhua Zhu, Bo Zhang, and Zheng Wang. 2020. An efficient approach for measuring semantic similarity combining WordNet and Wikipedia. *IEEE Access*, 8:184318–184338.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Isaiah Onando Mulang', Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. 2020. Evaluating the impact of knowledge graph context on entity disambiguation models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 2157–2160, New York, NY, USA. Association for Computing Machinery.

Jan Portisch, Michael Hladik, and Heiko Paulheim. 2021. Background knowledge in schema matching: Strategy vs. data. In *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings*, volume 12922 of *Lecture Notes in Computer Science*, pages 287–303. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2021. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Pavel Shvaiko and Jérôme Euzenat. 2011. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176.

Michael K. Smith, Chris Welty, and Deborah L. McGuinness. 2004. Owl web ontology language guide. https://www.w3.org/TR/2004/REC-owl-guide-20040210/.