Context-specific Adaptation of Subjective Content Descriptions

Felix Kuhr, Magnus Bender, Tanya Braun and Ralf Möller
University of Lübeck
Institute of Information Systems
Ratzeburgerallee 160, 23562 Lübeck
{kuhr,m.bender,braun,moeller}@ifis.uni-luebeck.de

***Abstract*—An agent in pursuit of a task may work with an individual collection of documents, which is known as a corpus. We assume that each document in the corpus is associated with additional location-specific data making the nearby content explicit by providing descriptions, references, or explanations about the content. Manually creating corpus- and location-specific data for documents is a time-consuming task. Thus, we are interested in using already existing data associated to documents in one corpus to enrich documents in another corpus without such data using the existing descriptions. This paper describes the problem for adapting location-specific data of documents in one corpus to documents in another corpus and presents an approach solving the problem. A case study shows the effectiveness of the adaptation approach.**

I. Introduction

An agent in pursuit of a task, explicitly or implicitly defined, may work with an individual collection of documents. From an agent-theoretic perspective, an agent is a rational, autonomous unit acting in a world perceived through sensors, fulfilling a defined task, e.g., providing document retrieval services given requests from users. We denote the collection of documents as a corpus and assume that a corpus represents a specific context, since collecting documents is not an end in itself. Documents in a given corpus might be associated with additional location-specific data making the content nearby the location explicit by providing descriptions, references, or explanations about the content at the location. We refer to these location-specific data as subjective content descriptions (SCDs). Kuhr et al. have introduced SCDs and have shown that the corpus specific SCD-word distribution, resulting from SCDs and text of documents, provides a value for different tasks of an agent in the context of a given corpus, e.g., classifying new documents to extend a corpus with documents from a specific category [6] or enriching documents with SCDs associated to other documents in the same corpus [14].

However, what can an agent do if presented with a new corpus containing documents having *no* associated SCDs? One technique to associate SCDs to documents having *no* associated SCDs is manually generating SCDs for those documents. But manually generating SCDs is a non-scalable and time-consuming task. So, the question is whether available SCDs associated to documents in a given (source) corpus provide a value for documents in another (target) corpus? Simply applying the source corpus specific SCD-word distribution to documents in a target corpus ignores the lexical difference across both corpora as well as the differing contexts of both corpora, e.g., the vocabulary of financial news articles differs from the vocabulary of biomedical research abstracts [3]. Thus, we are interested in adapting the SCD-word distribution of a source corpus to the documents in a target corpus while considering the lexical and context shift between both corpora.

In this work, we assume that documents in a target corpus containing only text can still indicate in which direction the target corpus differs from a source corpus so that we can take advantage of this indication while adapting the SCD-word distribution from the source corpus to documents in the target corpus. Specifically, the contributions of this paper are: (i) a definition of the context-specific SCD-word distribution adaptation problem, (ii) an approach to adjust the SCD-word distribution from a source corpus to a target corpus, and (iii) a case study on the effectiveness of the presented approach.

The remainder of this paper is structured as follows: We specify notations and recap SCDs as well as domain adaptation. Next, we present a new domain adaptation approach for SCDs including a case study. The paper ends with a look at related work, followed by a conclusion and future work.

II. Preliminaries

This section specifies notations, SCDs, and gives an overview of unsupervised domain adaptation.

A. Subjective Content Descriptions

SCDs are associated with locations in documents of a corpus and add value to the task of an agent, e.g., to optimize the performance of a document retrieval service. First, we formalize the setting of a corpus. Second, we define an SCD-word distribution for a corpus.

- A word w is a basic unit of discrete data from a vocabulary $\mathcal{V} = (w_1, \ldots, w_N)$, $N \in \mathbb{N}$, and can be represented as a one-hot vector of length N having a value of 1 where $w = w_i$ and 0's otherwise.
- A document d is represented by a sequence of $D \in \mathbb{N}$ words $(w_1^d, \ldots, w_D^d)$. Function $\#words(d)$ returns the total number of words in d from $\mathcal{V}$, i.e., D.
- A corpus $\mathcal{D}$ represents a set of $Z \in \mathbb{N}$ documents $\{d_1, \ldots, d_Z\}$ and $\mathcal{V}_\mathcal{D}$ returns the corpus-specific vocabulary $\mathcal{V}$ of $\mathcal{D}$.

- An SCD t can take any form. As such, its format may be highly diverse. A standardized format would be an Resource Description Framework (RDF) triple but, for our main contributions, the specific format is irrelevant.
- An SCD t can be associated to a position ρ in some document $d \in \mathcal{D}$. We use the term *located SCD* to describe an SCD t located at a position ρ and represent a located SCD by the tuple $\{(t, \{\rho_i\}_{i=1}^{l})\}$, where the set $\{\rho_i\}_{i=1}^{l}$ represents the $l \in \mathbb{N}$ positions in d that t is associated with.
- For each document $d \in \mathcal{D}$, there exists an *SCD set* $g(d)$ containing a set of k located SCDs $\{(t_j, \{\rho_i\}_{i=1}^{l_j})\}_{j=1}^{k}$.
- The set of all m SCDs in a corpus $\mathcal{D}$, ignoring the locations, is given by $\mathcal{T}_{\mathcal{D}_s} = \{t_j\}_{j=1}^{m}$
- For each located SCD t_j with position ρ in $g(d)$ exists a corresponding SCD window $win_{d,\rho}$ that refers to a sequence of words in d surrounding the position ρ in d, i.e., $win_{d,\rho} = (w^d_{(\rho-i)}, ..., w^d_{\rho}, ..., w^d_{(\rho+i)})$, $i \in \mathbb{N}$ and ρ marks the middle of the SCD window. The window-specific position of a word $w^d \in win_{d,\rho}$ is given by $pos(w^d, win_{d,\rho})$ (0-based numbering) and the size of window $win_{d,\rho}$ is given by $s(win_{d,\rho}) = 2i + 1$.
- Each word w^d in window $win_{d,\rho}$ around position ρ in document d is associated with an influence value $I(w^d, win_{d,\rho})$ representing the distance in the text between w^d and ρ. The closer w^d is positioned to ρ in $win_{d,\rho}$, the higher $I(w^d, win_{d,\rho})$ is. The influence value of w^d at $pos(w^d, win_{d,\rho})$ is distributed binomially, i.e., $I(w^d, win_{d,\rho}) = \binom{n}{k} \cdot q^k \cdot (1-q)^{n-k}$, where $n = s(win_{d',\rho})$, $k = pos(w^d, win_{d,\rho})$, and $q = \frac{\rho}{n}$.

Example 1 (Subjective Content Description)**.** *Let us assume that a document d starts with the following two sentences: "*I saved some money to buy a new mouse. The colour of the mouse is black*". Without additional information about the content, e.g., represented by an SCD associated to d, there are different interpretation possible about the content in d, e.g., a person bought a rodent, or a handheld device to move a cursor on a screen. If an SCD $t = (mouse, be, peripheral)$ is associated to the ninth word (*mouse*), t makes the content explicit by providing a description about the word* mouse*. Thus, an agent providing a document retrieval service benefits from t associated to d, since the SCD makes the content of d explicit and an agent can return d, e.g., after receiving a query like "Computer peripheral".*

B. SCD-word Distribution

We generate an additional representation for each of the m SCDs associated to documents in corpus $\mathcal{D}$ by building a vector of length n where $n = |\mathcal{V}_\mathcal{D}|$ s.t. each vector entry refers to a word $w \in \mathcal{V}_\mathcal{D}$. The entry itself is a probability describing how likely it is that a word occurs in an SCD window surrounding the position associated with the SCD, yielding an SCD-word distribution for each SCD. Algorithm 1 generates the SCD-word distribution for all m SCDs available in the SCD set $g(\mathcal{D})$. We represent the SCD-word distribution by an $m \times n$ matrix $\delta(\mathcal{D})$, where the SCD-word distribution vectors form the rows of the matrix:

Algorithm 1 Forming SCD-word distribution matrix $\delta(\mathcal{D})$

```
1:  function BUILDMATRIX(Corpus D)
2:      Input: D
3:      Output: δ(D)
4:      Initialize an m × n matrix δ(D) with zeros
5:      for each d ∈ D do
6:          for each t, ρ ∈ g(d) do
7:              for each w ∈ win_{d,ρ} do        ▷ Iterates over ρ
8:                  δ(D)[t][w] += I(w, win_{d,ρ})
9:          Normalize δ(D)[t]
10:     return δ(D)
```

$$\delta(\mathcal{D}) = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_m \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & w_2 & w_3 & \cdots & w_n \end{array} \\ \begin{pmatrix} v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{m,1} & v_{m,2} & v_{m,3} & \cdots & v_{m,n} \end{pmatrix} \end{array} \quad (1)$$

The input of Alg. 1 is a corpus $\mathcal{D}$ containing a set of documents associated with SCDs. In line 4, we instantiate an empty $\delta(\mathcal{D})$ by filling the matrix with zeros. Afterwards, we fill $\delta(\mathcal{D})$ based on the SCDs and words occurring in the documents of $\mathcal{D}$ using a maximum-likelihood strategy counting for each SCD t the number of occurrences of each word w in the corresponding windows $win_{d,\rho}$ of all documents in $\mathcal{D}$ and all positions. We weight the occurrences by the influence value of each word in a window (line 8). At the end of the outer loop, the SCD-word distribution of the current SCD t is normalized to yield a probability distribution for each SCD over the complete vocabulary, i.e., $p(\mathcal{V}_\mathcal{D} \mid t)$ (line 9). That is for each entry of a row (t) in $\delta(\mathcal{D})$, we divide the individual influence value by the sum of all influence values of that row. Please refer to [6] for details. Finally, Alg. 1 returns the SCD-word distribution matrix $\delta(\mathcal{D})$.

C. Unsupervised Domain Adaptation

For unsupervised domain adaptation, a set of labeled instances $\{(x_i, y_i)\}_{i=1}^{N^s}$ is given as a source, with N^s data points $x_i \in \mathcal{X}$, where $\mathcal{X}$ represents the feature space for observable data points, and corresponding labels y_i from a discrete set $\mathcal{Y}$. In discriminative models, each labeled instance (x_i, y_i) is drawn from an unknown joint distribution $p(x, y)$ but only $p(y \mid x)$ is available from the labeled instances. For the source, we can approximate the unknown joint distribution $p_s(x, y)$ using maximum likelihood estimation and parameters specific to the *source* domain. The goal is to estimate a distribution $p_t(x, y)$ for a target domain. However, the approximated distribution $p_s(x, y)$ will generally not work as $p_t(x, y)$.

Since a joint distribution $p(x, y)$ can be represented as

$$p(x, y) = p(y \mid x) \cdot p(x), \quad (2)$$

the mismatch between $p_s(x, y)$ and $p_t(x, y)$ arises because one of the two distributions $p(y \mid x)$ and $p(x)$ mismatches between source and target [12]. More specifically, the following two cases occur: (i) Target distribution $p_t(y \mid x)$ is similar to source distribution $p_s(y \mid x)$, but $p_t(x)$ deviates from $p_s(x)$. If $p_t(x)$ varies from $p_s(x)$, the source distribution $p_s(y \mid x)$ works only on those regions in the target domain where both $p_t(x, y)$ and $p_s(x, y)$ are similar. (ii) Target distribution $p_t(y \mid x)$ deviates from source distribution $p_s(y \mid x)$ in a way that $p_s(y \mid x)$ is no ideal estimator of $p_t(y \mid x)$.

The first case requires so-called *instance adaptation*, which depends on the data representation in source and target domain. The second case requires so-called *labeling adaptation*. There exist three ways for labeling adaptation: (i) changing the representation of instances, (ii) adapting a prior in the source domain to adjust the source domain to the given target domain, and (iii) pruning instances in the source domain.

Next, we present instance adaptation and labeling adaptation (based on pruning) for the SCD-setting.

III. Adapting Subjective Content Descriptions

In the unsupervised domain adaptation setting for SCDs, we assume that a source corpus $\mathcal{D}_s$ containing documents associated with located SCDs and a target corpus $\mathcal{D}_t$ without such SCDs are available. We are interested in associating SCDs from documents in $\mathcal{D}_s$ to documents in $\mathcal{D}_t$ since manually generating new SCDs for documents in $\mathcal{D}_t$, considering the context of $\mathcal{D}_t$, is a time-consuming task. Since, in our setting, SCDs generate words, an SCD-word distribution $\delta(\mathcal{D}_t)$ for $\mathcal{D}_t$ would allow us to estimate most probably suited SCDs (MPSCDs). As we can only form $\delta(\mathcal{D}_s)$, we are interested in adapting $\delta(\mathcal{D}_s)$ to $\mathcal{D}_t$ to estimate MPSCDs. Next, we define domain adaptation in the form of an instance and an labelling adaptation problem and then present an approach to solving each of the problems.

A. Domain Adaptation Problem

The generative model of SCDs allows for estimating the full joint distribution of words and SCDs $\gamma(\mathcal{V}_{\mathcal{D}_S}, \mathcal{T}_{\mathcal{D}_s})$ for the source $\mathcal{D}_s$, $\gamma(\mathcal{D}_s)$ for short:

$$\gamma(\mathcal{V}_{\mathcal{D}_S}, \mathcal{T}_{\mathcal{D}_s}) = p(\mathcal{V}_{\mathcal{D}_s} \mid \mathcal{T}_{\mathcal{D}_s}) \cdot p(\mathcal{T}_{\mathcal{D}_s}) = \delta(\mathcal{D}_s) \cdot p(\mathcal{T}_{\mathcal{D}_s}), \tag{3}$$

where we assume uniform prior distribution $p(\mathcal{T}_{\mathcal{D}_s})$. As mentioned above, we do not have $\delta(\mathcal{D}_t)$ for the target $\mathcal{D}_t$ as we do not have SCDs and only if $\gamma(\mathcal{D}_s) = \gamma(\mathcal{D}_t)$, we can use $\gamma(\mathcal{D}_s)$ or more specifically, $\delta(\mathcal{D}_s)$, directly to estimate valuable SCDs for documents in $\mathcal{D}_t$.

What we do have from $\mathcal{D}_t$ are the words, meaning we can estimate a prior distribution over the vocabulary using maximum likelihood estimation, i.e., counting the occurrences of words and normalising the counts. Therefore, instead of factorizing $p(x, y)$ as given in Eq. (2), we factorize $p(x, y)$ the other way around:

$$p(x, y) = p(x \mid y)p(y), \tag{4}$$

which yields for our setting:

$$\gamma(\mathcal{V}_{\mathcal{D}}, \mathcal{T}_{\mathcal{D}}) = p(\mathcal{T}_{\mathcal{D}} \mid \mathcal{V}_{\mathcal{D}}) \cdot p(\mathcal{V}_{\mathcal{D}}). \tag{5}$$

The distribution $p(\mathcal{V}_{\mathcal{D}_s})$ is the prior over the vocabulary in $\mathcal{D}$. Given our assumption of a uniform prior distribution in Eq. (3), we can then calculate $p(\mathcal{T}_{\mathcal{D}_s} \mid \mathcal{V}_{\mathcal{D}_s})$ given $p(\mathcal{V}_{\mathcal{D}_s})$ and $\delta(\mathcal{D}_s)$ for our source corpus as the following holds given Eqs. (3) and (5):

$$\delta(\mathcal{D}_s) = \gamma(\mathcal{D}_s) = p(\mathcal{T}_{\mathcal{D}_s} \mid \mathcal{V}_{\mathcal{D}_s}) \cdot p(\mathcal{V}_{\mathcal{D}_s}). \tag{6}$$

This setup allows us to define instance and labelling adaptation problems for SCDs to adapt the distributions of $\mathcal{D}_s$ to $\mathcal{D}_t$.

Problem 1 (Instance Adaptation)**.** *The target distribution $p(V_{\mathcal{D}_t})$ deviates from source distribution $p(V_{\mathcal{D}_s})$.*

Problem 1 can occur if there is a lexical gap between $\mathcal{D}_s$ and $\mathcal{D}_t$, e.g., one corpus contains academic articles and the other corpus contains newspaper articles about the same subject. As we can estimate both vocabulary distributions using maximum likelihood estimation, we can incorporate their difference into $\delta(\mathcal{D}_s)$ to estimate a $\hat{\delta}(\mathcal{D}_s)$ for $\mathcal{D}_t$.

Problem 2 (Labeling Adaptation)**.** *The target distribution $\delta(\mathcal{D}_t)$ deviates from the source distribution $\delta(\mathcal{D}_s)$.*

Problem 2 can occur if the contextual difference between documents in $\mathcal{D}_s$ and $\mathcal{D}_t$ is large, e.g., the content of documents between both corpora differs. We define Problem 2 directly on $\delta(\mathcal{D})$ and not on $p(\mathcal{T}_{\mathcal{D}} \mid \mathcal{V}_{\mathcal{D}})$ since the vocabulary distributions are considered identical between source and target corpus, making the switch between $\delta(\mathcal{D})$ and $p(\mathcal{T}_{\mathcal{D}} \mid \mathcal{V}_{\mathcal{D}})$ without further consequence if looking to adapt the distributions beyond the vocabulary, which allows us to tackle the problem concentrating on $\delta(\mathcal{D}_s)$.

The goal after adaptation is to automatically enrich documents in $\mathcal{D}_t$ with SCDs associated to documents in $\mathcal{D}_s$ using an adapted $\hat{\delta}(\mathcal{D}_s)$. Next, we present domain adaptation techniques for both problems before recapping how to estimate MPSCDs to enrich $\mathcal{D}_t$.

B. Instance Adaptation

We assume that the word distribution $p(V_{\mathcal{D}_t})$ is different from the word distribution $p(V_{\mathcal{D}_s})$. Generally, we can approximate $p(V_{\mathcal{D}_s})$ and $p(V_{\mathcal{D}_t})$ of $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively, s.t. we can adapt $\delta(\mathcal{D}_s)$ based on the difference between $p(V_{\mathcal{D}_s})$ and $p(V_{\mathcal{D}_t})$, which results in an adapted version $\hat{\delta}(\mathcal{D}_s)$, which becomes $\delta(\mathcal{D}_t)$ that is optimized for documents in $\mathcal{D}_t$. We introduce an instance adaptation approach to adapt the value of each word in $\delta(\mathcal{D}_s)$ based on the difference in word frequencies between both corpora. Algorithm 2 describes the adaptation approach in detail, calculating for both corpora the corresponding word frequency vectors f_s and f_t to estimate the word distributions $p(V_{\mathcal{D}_s})$ and $p(V_{\mathcal{D}_t})$ for $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively (lines 4-5 and function CountFreq). Afterwards, the algorithm estimates the difference between both word distributions to obtain an instance adaptation vector v (lines 6-8). The weighting factor (WF) depends on the lexical gap

Algorithm 2 Instance Adaptation by Instance Weighting

```
1: function InstanceWeighting(D_s, D_t)
2:     v ← new zero-vector of length |V(D_s)|
3:     δ̂(D_s) ← δ(D_s)
4:     f_s ← countFreq(D_s)
5:     f_t ← countFreq(D_t)
6:     for each w ∈ V_{D_s} do                    ▷ Calculate weights
7:         if w ∈ V(D_t) then
8:             v[w] ← f_s[w] · (1 − (f_s[w] − f_t[w])) · WF
9:     for each row t in δ̂(D_s) do                ▷ Reweight δ(D_t)
10:        c ← 0
11:        for each column w in δ̂(D_s) do
12:            δ̂(D_s)[t][w] ← δ̂(D_s)[t][w] · v[w]
13:            c ← c + δ̂(D_s)[t][w]
14:        δ̂(D_s)[t] ← (1/c) · δ̂(D_s)[t]          ▷ Normalize
15:    return δ̂(D_t)
16: function countFreq(D)
17:    f ← new zero-vector of length |V(D_s) ∪ V(D_t)|
18:    c ← 0
19:    for each d ∈ D do
20:        c ← c + #words(d)
21:        for each w ∈ d do
22:            f[w] ← f[w] + 1
23:    return (1/c) · f                           ▷ Normalize
```

between $p(V_{\mathcal{D}_s})$ and $p(V_{\mathcal{D}_t})$. Next, Alg. 2 adapts the entries of each row of $\delta(\mathcal{D}_s)$ by multiplying the entry with the corresponding entry in v (lines 9-14), which results in an adapted distribution $\hat{\delta}(\mathcal{D}_s)$ used as $\delta(\mathcal{D}_t)$. Next, we argue that Alg. 2 solves Problem 1.

Theorem 1. *Algorithm 2 solves Problem 1.*

Proof sketch. Since the difference between word distributions characterizes Problem 1 lies in and the words of both corpora are available, we estimate both word distributions using maximum likelihood estimation and adapt $\delta(\mathcal{D}_s)$ according to the difference between the two distributions (see line 8), leading to an SCD-word distribution $\delta(\mathcal{D}_t)$ for the target corpus using Eq. (6), which solves Problem 1. □

Next, we provide an approach to tackle Problem 2.

C. Labeling Adaptation

The joint probability distribution $\gamma(\mathcal{D}_s)$ is based on word distribution $p(V_{\mathcal{D}_s})$ and word-SCD distribution $p(\mathcal{T}_{\mathcal{D}} \mid \mathcal{V}_{\mathcal{D}})$. Even if the documents in both corpora share the same word distribution, the SCD-word distribution of $\mathcal{D}_s$ and $\mathcal{D}_t$ are different because of the difference in $p(\mathcal{T}_{\mathcal{D}} \mid \mathcal{V}_{\mathcal{D}})$.

To estimate where and why $\delta(\mathcal{D}_t)$ differs from $\delta(\mathcal{D}_s)$, since it is not the vocabulary, we would need some prior about the data in the target corpus. However, the only available data from $\mathcal{D}_t$ are the words in the documents themselves, since no SCDs are associated to documents in $\mathcal{D}_t$. Comparing documents from the source corpus with documents from the target corpus requires a common ground between documents in both corpora. Generally, we can use any word-based document representation to compare documents. A frequently used text-mining technique to represent documents is given by a *topic model*, which is a statistical model for discovering *topics*, or hidden semantic structures, in a collection of documents. Thus, we generate for both corpora $\mathcal{D}_s$ and $\mathcal{D}_t$ a topic model using the well-known latent Dirichlet allocation (LDA) approach [2] and refer to the topic models for $\mathcal{D}_s$ and $\mathcal{D}_t$ by $M(\mathcal{D}_s)$ and $M(\mathcal{D}_t)$, respectively. With a topic model $M(\mathcal{D})$, we can represent and compare two documents $d_i \in \mathcal{D}$ and $d_j \in \mathcal{D}$ by their topic distributions θ_{d_i} and θ_{d_j}, e.g., by calculating the Hellinger distance [11] between both distributions over K topics and is defined as follows:

$$H(\theta_{d_i}, \theta_{d_j}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{K} \left(\sqrt{\theta_{d_i,k}} - \sqrt{\theta_{d_j,k}} \right)^2}$$

Based on $M(\mathcal{D}_s)$ and $M(\mathcal{D}_t)$, we can determine documents in $\mathcal{D}_s$ having a different topic distribution to documents in $\mathcal{D}_t$ and remove these documents from $\mathcal{D}_s$ if the Hellinger distance is higher than a defined threshold τ. This process yields to a pruned corpus $\mathcal{D}_{s'}$. For $\mathcal{D}_{s'}$, we can generate $\delta(\mathcal{D}_{s'})$ directly that is optimized for documents in $\mathcal{D}_t$.

Algorithm 3 Labeling Adaptation by Instance Pruning

```
1: function InstancePruning(D_s, D_t)
2:     Generate topic models M(D_s), M(D_t) with θ_{D_s}, θ_{D_t}
3:     Identify topic mapping σ between M(D_s) and M(D_t)
4:     C_{D_t} ← ∅
5:     for each d_t ∈ D_t do
6:         for each d_s ∈ D_s do
7:             if H_σ(θ_{d_t}, θ_{d_s}) < τ then
8:                 C_{D_t} ← C_{D_t} ∪ d_s
9:     Build δ(C_{D_t})                           ▷ See Alg. 1
10:    return δ(C_{D_t})
```

Algorithm 3 describes the document pruning approach in detail. Since each corpus represents a specific context, Alg. 3 generates for source corpus and target corpus a topic model $M(\mathcal{D}_s)$ and $M(\mathcal{D}_t)$, respectively (line 2). We cannot directly compare the topic distribution from a document of $\mathcal{D}_s$ with the topic distribution from a document of $\mathcal{D}_t$ because both topic distributions are generated from different models s.t. the first topic of $M(\mathcal{D}_s)$ is not the first topic in $M(\mathcal{D}_t)$. Thus, we need a mapping σ between the topics generated from $M(\mathcal{D}_s)$ and $M(\mathcal{D}_t)$, e.g., by analyzing the topic coherence [5], which is referenced in line 3. Next, Alg. 3 generates a cluster $\mathcal{C}_{D_t}$ containing only those documents from source corpus $\mathcal{D}_s$ having a topic similarity with documents in the target corpus, in other words having a Hellinger distance smaller than threshold τ. Algorithm 3 determines the similarity between two documents based on the Hellinger distance of their topic distributions. To calculate the Hellinger distance, we need σ to align the topics of both distributions. If the distance between

document-topic distribution θ_{d_s} and θ_{d_t} is below τ, we add $d_s \in \mathcal{D}_s$ to $\mathcal{C}_{D_t}$ (line 5-8). Afterwards, Alg. 3 generates an SCD-word distribution $\delta(\mathcal{C}_{\mathcal{D}_t})$ for $\mathcal{C}_{\mathcal{D}_t}$ s.t. $\delta(\mathcal{C}_{\mathcal{D}_t})$ is optimized for the target corpus $\mathcal{D}_t$. Next, we argue that Algorithm 3 solves Problem 2.

Theorem 2. *Algorithm 3 solves Problem 2.*

Proof sketch. Problem 2 posits that the source distribution $\delta(\mathcal{D}_s)$ deviates from the target distribution $\delta(\mathcal{D}_t)$. Two main reasons for $\delta(\mathcal{D}_s)$ not working for $\mathcal{D}_t$ lie in (i) $\delta(\mathcal{D}_s)$ referencing SCDs not relevant for $\mathcal{D}_t$ or (ii) SCDs being associated with different words in the context of $\mathcal{D}_t$. With Alg. 3 working only with a subset of $\mathcal{D}_s$, there are two possible effects, when comparing $\delta(\mathcal{D}_{s'})$ with $\delta(\mathcal{D}_s)$: (i) Pruned documents contain SCDs that are also associated to documents in the reduced corpus $\mathcal{D}_{s'}$, resulting in the same set of SCDs in SCD-word distribution $\delta(\mathcal{D}_{s'})$, but different word vector entries. (ii) Pruned documents contain SCDs not associated to other documents resulting in a reduced set of SCDs in $\delta(\mathcal{D}_{s'})$. The effects counteract the two main reasons for $\delta(\mathcal{D}_s)$ not applying to $\delta(\mathcal{D}_t)$. As such, instead of using $\delta(\mathcal{D}_s)$, we use $\delta(\mathcal{D}_{s'}) = \delta(\mathcal{C}_{D_t})$ for $\delta(\mathcal{D}_t)$, solving Problem 2. □

With Algs. 2 and 3, we have two approaches to solve Problems 1 and 2, respectively, by adapting $\delta(\mathcal{D}_s)$ to $\mathcal{D}_t$, yielding an adapted $\hat{\delta}(\mathcal{D}_s)$ for $\delta(\mathcal{D}_t)$. Based on this $\hat{\delta}(\mathcal{D}_s)$, we can now estimate MPSCDs for $\mathcal{D}_t$, automatically enriching $\mathcal{D}_t$ with SCDs associated to documents in $\mathcal{D}_s$.

D. Estimating MPSCDs

The idea behind MPSCDs is to find those SCDs that are most likely to have generated the words in a particular window. Therefore, given $\delta(\mathcal{D})$ and a document without SCDs d, one can slide a tumbling window over the words in d and estimate the SCDs that fit the words in the windows best.

Algorithm 4 outlines estimating MPSCDs, which we do for each document $d \in \mathcal{D}_t$. The input parameters in the domain adaptation scenario are: (i) a document $d \in \mathcal{D}_t$, (ii) the number of MPSCDs M we are interested in estimating for d, and (iii) the (adapted) SCD-word distribution $\hat{\delta}(\mathcal{D}_s)$. In line 6-7, we build a vector representation $\delta(win_{d,\rho})$ for each of the M windows in d to estimate the SCD from $\hat{\delta}(\mathcal{D}_s)$ that has a vector representation most similar to $\delta(win_{d,\rho})$. The SCD t that is most similar to $\delta(win_{d,\rho})$ is given by the cosine similarity (line 8). The output of Alg. 4 is an SCD set $g(d)$ containing the M MPSCDs and $\mathcal{W}$ containing the similarity values of the M MPSCDs in $g(d)$.

Next, we present a case study showing how both approaches perform on four data sets and to discuss under which circumstances which approach works best.

IV. Case Study

After having introduced unsupervised domain adaptation techniques for SCDs, we present a case study analysing the performance of both adaptation techniques in estimating SCDs for documents in a target corpus using only the SCDs associated to documents in a given source corpus.

Algorithm 4 Estimating MPSCDs

1: **function** ESTIMATEMPSCD(d, M, $\hat{\delta}(\mathcal{D}_s)$)
2: $\quad\sigma \leftarrow \frac{words(d)}{M}$, $\rho \leftarrow \frac{\sigma}{2}$, $\mathcal{W} \leftarrow \emptyset$
3: $\quad$**for** $\rho \leftarrow \frac{\sigma}{2}$; $\rho \leq words(d)$; $\rho += \sigma$ **do**
4: $\quad\quad$Set up $win_{d,t,\rho}$ of size σ around ρ with $t = \perp$
5: $\quad\quad\hat{\delta}(win_{d,t,\rho}) \leftarrow$ new zero-vector of length n
6: $\quad\quad$**for** $w \in win_{d,t,\rho}$ **do**
7: $\quad\quad\quad\hat{\delta}(win_{d,t,\rho})[w] += I(w, win_{d,t,\rho})$
8: $\quad\quad t \leftarrow \arg\max_{t_i} \frac{\hat{\delta}(\mathcal{D}_s)[i] \cdot \hat{\delta}(win_{d,t,\rho})}{|\hat{\delta}(\mathcal{D}_s)[i]| \cdot |\hat{\delta}(win_{d,t,\rho})|}$ in $win_{d,t,\rho}$
9: $\quad\quad sim \leftarrow \max_{t_i} \frac{\hat{\delta}(\mathcal{D}_s)[i] \cdot \hat{\delta}(win_{d,t,\rho})}{|\hat{\delta}(\mathcal{D}_s)[i]| \cdot |\hat{\delta}(win_{d,t,\rho})|}$
10: $\quad\quad\mathcal{W} \leftarrow \mathcal{W} \cup \{(sim, win_{d,t,\rho})\}$
11: $\quad\quad g(d) \leftarrow g(d) \cup \{t\}$
12: $\quad$**return** $g(d)$, $\mathcal{W}$

We have implemented Alg. 2 and Alg. 3 as a Java program to analyze the performance of both adaptation techniques by estimating the MPSCDs for documents in the target corpus using Alg. 4 before and after adapting the SCD-word distribution $\delta(\mathcal{D}_s)$ of a source corpus $\mathcal{D}_s$ for documents in $\mathcal{D}_t$.We describe the data sets, necessary preprocessing techniques, and the evaluation workflow, and present the results of both domain adaptation techniques.

A. Data Sets

We have selected articles out of the open and widely accessible online encyclopedia *Wikipedia* to make our experiments reproducible. The data sets contain two sets of articles, which have been grouped by Wikipedia, representing the specific context of a corpus. The SCDs associated to documents in one set represent possibly valuable descriptions for documents in the other set. In the first data set, we use documents about presidents of the United States of America between 1789 and 2017[1] and documents about prime ministers of the United Kingdom between 1721 and 2019[2] as source and target corpus, respectively. In the second data set, we use documents about cities in the United States of America[3] and cities in Europe[4]. The number of documents in the source and target corpus is similar for both data sets.

There are no SCDs associated to Wikipedia articles, because Wikipedia is an online encyclopedia trying to provide objective reference work instead of being a personal reference library containing documents about a specific context. Thus, we have to associate SCDs to documents from the source and target corpus to evaluate the performance of our unsupervised domain adaptation techniques. As stated earlier, SCDs add additional data to documents, making the content explicitly by providing descriptions, references, or explanations about the content. So, we extract data from the text of the Wikipedia articles using Stanford OpenIE [1]. OpenIE tools extract

[1]US president data set - https://bit.ly/2Z1v1G9
[2]UK prime ministers data set - https://bit.ly/3iKbN2W
[3]US cities - https://bit.ly/3jUua5H
[4]European cities - https://bit.ly/34WXMsE

relation tuples (RDF triples) directly from the plain text of an article and associate the tuples to the position in the document they have been extracted from. As such, the relational tuples act as located SCDs for the documents in this case study.

B. Evaluation Setup

Given each data set, we choose one article set to be the source corpus $\mathcal{D}_s$ and one article set to be the target corpus $\mathcal{D}_t$. We use the located SCDs associated to documents in $\mathcal{D}_t$ as ground truth to evaluate the performance of domain adaptation. We concentrate on those SCDs associated to documents in $\mathcal{D}_t$, which also appear in $\mathcal{D}_s$, as only those can be correctly associated by using $\delta(\mathcal{D}_s)$ or any adapted version of $\delta(\mathcal{D}_s)$. For the evaluation, we then remove all SCDs associated to documents in $\mathcal{D}_t$ and use the adaptation techniques to adapt the SCD-word distribution $\delta(\mathcal{D}_s)$ of $\mathcal{D}_s$ to estimate the SCD for documents in $\mathcal{D}_t$ and compare the result with the ground-truth SCDs. Specifically, we evaluate the adaptation performance of Alg. 2 and Alg. 3 by comparing the estimated MPSCDs for documents in the target corpus before and after adapting the SCD-word distribution $\delta(\mathcal{D}_s)$ using the positive predictive value (PPV). The PPV is defined by

$$PPV = \frac{tp}{tp + fp}$$

True positives (tp) refer to the number of SCDs that has been correctly estimated and false positives (fp) refer to the number of SCDs that has been falsely estimated.

We consider the following four cases using $\delta(\mathcal{D}_s)$ as well as different adapted versions of $\delta(\mathcal{D}_s)$:

(i) Baseline: Using $\delta(\mathcal{D}_s)$ without any adaptation for documents in $\mathcal{D}_t$.
(ii) Instance adaptation: Using Alg. 2 reweighing the influence values in $\delta(\mathcal{D}_s)$ based on the words in $\mathcal{D}_t$ using the best weighting factors identified before.
(iii) Labeling adaptation: Using Alg. 3 selecting only document from $\mathcal{D}_s$ having a high topic similarity with documents in $\mathcal{D}_t$ to generate a target corpus optimized SCD-word distribution.
(iv) Both: Applying both algorithms in sequence, since both Problems 1 and 2 can occur simultaneously.

C. Evaluation Workflow

We download all necessary documents from Wikipedia using a Python script and the Wikipedia API and store the documents in the respective corpus. Afterwards, we preprocess the documents by performing the following tasks: (i) lowercase all characters, (ii) stem the words, (iii) tokenize the result, (iv) eliminate tokens from a stop-word list containing 337 words, and (v) extract relation tuples using OpenIE. The first four tasks are standard preprocessing tasks in the NLP community, transforming the text of documents into more digestible form for machine learning algorithms, to increase their performance [19].

For each data set, the preprocessing steps result in a source and target corpus containing documents that are associated with located SCDs. Then, we evaluate the adaptation performance of both unsupervised domain adaptation techniques by performing the following tasks for each data set:

(i) Identify the SCDs occurring in both corpora to determine the set of SCDs we can correctly associate to documents in target corpus $\mathcal{D}_t$ using the (adapted) SCD-word distribution $\hat{\delta}(\mathcal{D}_s)$ of source corpus $\mathcal{D}_s$.
(ii) Remove all SCDs associated to documents in $\mathcal{D}_t$ s.t. $\mathcal{D}_t$ represents a common reference library, where documents contain only text and no SCDs.
(iii) Calculate $\delta(\mathcal{D}_s)$ using Alg. 1.
(iv) Estimate MPSCDs for documents in $\mathcal{D}_t$ using Alg. 4 with the original SCD-word distribution $\delta(\mathcal{D}_s)$.
(v) Calculate the baseline PPV for the SCDs of documents in $\mathcal{D}_t$ using the original SCD-word distribution $\delta(\mathcal{D}_s)$ s.t. we can compare the performance of both domain adaptation techniques with the baseline PPV.
(vi) Perform instance adaptation (Alg. 2) and labeling adaptation (Alg. 3) on the source corpus and use the adapted versions of $\delta(\mathcal{D}_s)$ to estimate MPSCDs for documents in $\mathcal{D}_t$ using Alg. 4.
(vii) Calculate the PPVs for SCDs associated to documents in $\mathcal{D}_t$ after performing instance and labeling adaptation and compare the performance of both adaptation techniques with the baseline PPV.

D. Results

This section presents results regarding the weighting factors as well as the domain adaptation approaches.

a) Weighting Factor: We show the effect of different weighting factors for both data sets in Fig. 1. For data set 1, higher weighting factors lead to higher PPV. For data set 2, smaller weighting factors lead to higher PPV. As we have expected, the PPV for data set 2 is higher than for data set 1 using only Alg. 2.

b) Domain Adaption: Figure 2 presents the performance of the four cases described in Section IV-B using the source corpus specific SCD-word distribution $\delta(\mathcal{D}_s)$ and three adapted versions of $\delta(\mathcal{D}_s)$.

We evaluate for both data sets the performance of estimating the MPSCDs for documents in the target corpus considering each of the four cases. Algorithm 4 selects the MPSCD based on the similarity value of SCDs. The similarity value of the first k MPSCDs might be almost the same. Thus, we consider the top-k MPSCDs and mark an estimated MPSCD as true positive, if the estimated SCD is in the top-k MPSCDs. We use three different settings considering the top-1, top-5 and top-10 MPSCDs, represented by PPV@1, PPV@5, and PPV@10, respectively. In case of labeling adaptation, we use 15 topics for the topic model, since the topic model containing 15 topics has the best quality w.r.t the perplexity of the models and use $\tau = 0.6$ (Alg. 3 line 7) as threshold to decide if two documents are similar.

As we have expected, the performance using the original SCD-word distribution $\delta(\mathcal{D}_s)$ is low for data set 1, because

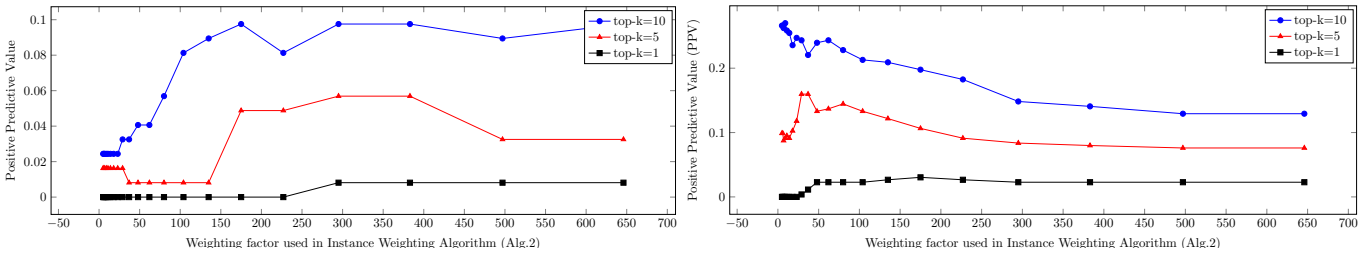

Fig. 1: Estimating best weighting factors for data set 1 (left) and data set 2 (right) used in Algorithm 2.

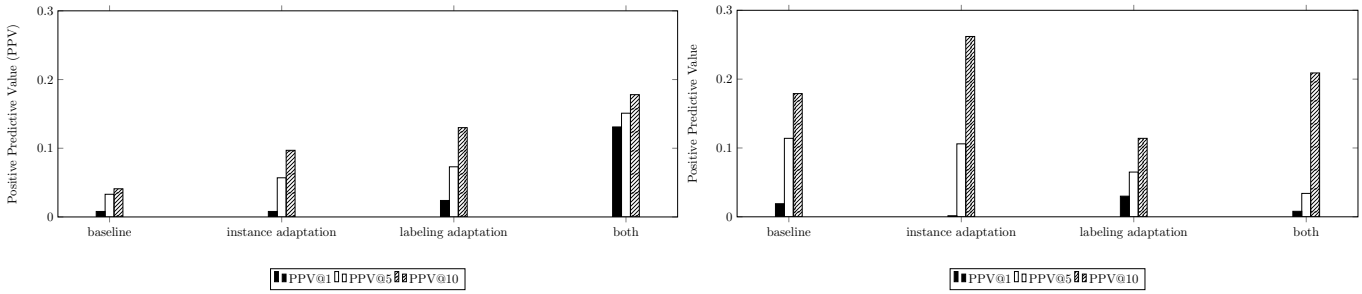

Fig. 2: Performance of no adaptation, instance adaptation, labeling adaptation, and a combination of both techniques using first labeling followed by instance adaptation (both) for data set 1 (left) and data set 2 (right). We use PPV@1, PPV@5, and PPV@10, to represent the PPV performance looking at the top-k = 1,5,10 MPSCDs, respectively.

of the varying context and lexical difference between documents in source and target corpus. Instance adaptation slightly increases the PPV in comparison to the baseline. Pruning documents in the source corpus by Alg. 3 results in even better PPVs. However, the combination of both adaptation techniques leads to best results and the different between PPV@1 and PPV@10 is remarkably small.

For the second data set, the PPV using the original SCD-word distribution $\delta(\mathcal{D}_s)$ is clearly higher than for data set 1, since the vocabulary used in source and target corpus for data set 2 is more similar than for data set 1. Optimizing $\delta(\mathcal{D}_s)$ using instance adaptation results in best PPV performance considering the top-10 MPSCDs. Interestingly, the PPV decreases by performing labeling adaptation. One reason for the decreasing performance is given by pruning some documents in the source corpus containing valuable SCDs for documents in the target corpus.

V. Related Work

In this section, we position our work in the field of domain adaptation, where the source data distribution is different (but related) to the target data distribution and the task of an agent is the same for both domains. In the last decades, the interest in unsupervised domain adaptation has increased, which is the task of modifying a model trained on labeled data available in a source domain to obtain better performance on data available in a target domain, without having labeled data in the target domain. The adaptation between corpora is the most common setting in the NLP community [15], and different challenges exist, e.g., having (i) a different representation for same entities, (ii) a difference in the context and the vocabulary of documents, (iii) a difference in word-sense distribution. Different forms of domain adaptation have driven progress for various tasks in named-entity recognition (NER) [3, 8], automatic capitalization [4], word-sense disambiguation [13], and part-of-speech tagging [16].

Many unsupervised domain adaptation methods focus on feature distribution matching between the source and the target domains by (i) reweighting or selecting samples from the source corpus [8, 9, 17], (ii) performing feature space transformation by mapping source distribution to the target distribution [10], and (iii) modifying the feature representation itself instead of reweighting or selecting samples from the source [7]. This work contributes to the first method by providing an approach to reuse SCDs that are associated to documents in a source corpus for documents in a target corpus. Our approach aims at matching the feature space distribution from the source domain to the target domain s.t. we can directly use the adapted SCDs-word distribution to estimate the MPSCDs for documents in a target corpus.

To the best of our knowledge, there exists no domain adaptation approach focussing on the task of context-specific SCDs for unlabeled documents in a target corpus.

VI. Conclusion and Outlook

If an agent finds a new corpus containing documents with no associated SCDs, the contributions of this paper enable the agent to adapt SCDs associated to documents in its library to documents in the new corpus s.t. the agent does not need to create new SCDs manually and can directly use the SCDs to, e.g., identify similar documents based on SCDs. Specifically, we define the problem of SCD adaptation

between corpora and present instance adaptation and labeling adaptation techniques for location specific content descriptions considering the corpus-specific context and vocabulary used in the documents in both corpora. Whereas instance adaptation is based on adapting varying word distributions, labeling adaptation requires adapting a complete SCD-word distribution, which we base on topic similarity between documents of both corpora. Performing domain adaptation on SCDs yields to an initial set of valuable SCDs supporting an agent working with documents in a new corpus.

In our future work, we are interested in a domain adaptation approach based on a parameterized model such that the documents in a target corpus influence labeling function parameters that optimize adapting the SCD-word distribution from the source corpus to the target corpus.

References

[1] Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. "Leveraging linguistic structure for open domain information extraction". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. 2015.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022.

[3] John Blitzer, Ryan McDonald, and Fernando Pereira. "Domain adaptation with structural correspondence learning". In: *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2006, pp. 120–128.

[4] Ciprian Chelba and Alex Acero. "Adaptation of maximum entropy capitalizer: Little data can help a lot". In: *Computer Speech & Language* 20.4 (2006), pp. 382–399.

[5] Felix Kuhr and Magnus Bender and Tanya Braun and Ralf Möller. "Maintaining Topic Models for Growing Corpora". In: *IEEE 14th International Conference on Semantic Computing, ICSC 2020, San Diego, CA, USA, February 3-5, 2020*. 2020, pp. 451–458.

[6] Felix Kuhr and Tanya Braun and Magnus Bender and Ralf Möller. "To Extend or not to Extend? Context-specific Corpus Enrichment". In: *Proceedings of AI 2019: Advances in Artificial Intelligence*. Springer, 2019.

[7] Yaroslav Ganin and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[8] Boqing Gong, Kristen Grauman, and Fei Sha. "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation". In: *International Conference on Machine Learning*. 2013, pp. 222–230.

[9] Gong, Boqing and Shi, Yuan and Sha, Fei and Grauman, Kristen. "Geodesic flow kernel for unsupervised domain adaptation". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2066–2073.

[10] Gretton, Arthur and Borgwardt, Karsten and Rasch, Malte and Schölkopf, Bernhard and Smola, Alex J. "A kernel method for the two-sample-problem". In: *Advances in neural information processing systems*. 2007, pp. 513–520.

[11] Ernst Hellinger. "Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen." In: *Journal für die reine und angewandte Mathematik* 136 (1909), pp. 210–271.

[12] Jing Jiang and ChengXiang Zhai. "Instance weighting for domain adaptation in NLP". In: *Proceedings of the 45th annual meeting of the association of computational linguistics*. 2007, pp. 264–271.

[13] Komiya, Kanako and Suzuki, Shota and Sasaki, Minoru and Shinnou, Hiroyuki and Okumura, Manabu. "Domain Adaptation for Word Sense Disambiguation Using Word Embeddings". In: *International Conference on Computational Linguistics and Intelligent Text Processing*. Springer. 2017, pp. 195–206.

[14] Felix Kuhr, Bjarne Witten, and Ralf Möller. "On Corpus-driven Annotation Enrichment". In: *13th IEEE International Conference on Semantic Computing*. IEEE Computer Society, 2019.

[15] Qi Li. "Literature survey: domain adaptation algorithms for natural language processing". In: *Department of Computer Science The Graduate Center, The City University of New York* (2012), pp. 8–10.

[16] Luisa März, Dietrich Trautmann, and Benjamin Roth. "Domain adaptation for part-of-speech tagging of noisy user-generated text". In: *arXiv preprint arXiv:1905.08920* (2019).

[17] Viswa Mani Kiran Peddinti and Prakriti Chintalapoodi. "Domain adaptation in sentiment analysis of twitter". In: *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*. 2011.

[18] Anders Søgaard. "Semi-supervised learning and domain adaptation in natural language processing". In: *Synthesis Lectures on Human Language Technologies* 6.2 (2013), pp. 1–103.

[19] S Vijayarani, Ms J Ilamathi, and Ms Nithya. "Preprocessing techniques for text mining-an overview". In: *International Journal of Computer Science & Communication Networks* 5.1 (2015), pp. 7–16.