Searching for Optimal Subword Tokenization in Cross-domain NER

Ruotian Ma†*, Yiding Tan†*, Xin Zhou†, Xuanting Chen†, Di Liang‡, Sirui Wang‡, Wei Wu‡, Tao Gui†.†

1School of Computer Science, Fudan University, Shanghai, China
2Institute of Modern Languages and Linguistics, Fudan University, Shanghai, China
3Meituan Inc., Beijing, China
{rtma19,yidingtan20,tgui}@fudan.edu.cn

Abstract

Input distribution shift is one of the vital problems in unsupervised domain adaptation (UDA). The most popular UDA approaches focus on domain-invariant representation learning, trying to align the features from different domains into similar feature distributions. However, these approaches ignore the direct alignment of input word distributions between domains, which is a vital factor in word-level classification tasks such as cross-domain NER. In this work, we shed new light on cross-domain NER by introducing a subword-level solution, X-Piece, for input word-level distribution shift in NER. Specifically, we re-tokenize the input words of the source domain to approach the target subword distribution, which is formulated and solved as an optimal transport problem. As this approach focuses on the input level, it can also be combined with previous DIRL methods for further improvement. Experimental results show the effectiveness of the proposed method based on BERT-tagger on four benchmark NER datasets. Also, the proposed method is proved to benefit DIRL methods such as DANN.

1 Introduction

Unsupervised domain adaptation (UDA) is a widely concerned problem in Machine Learning [Pan et al., 2010; Ganin et al., 2016], which is also a significant point in Named Entity Recognition (NER) [Chen and Moschitti, 2019]. The main concern in UDA is to perform a given task on a target domain with only unlabeled data provided, while having rich labeled data on a source domain. A key problem is that the data distribution varies across different domains, resulting in poor performance when directly adopting a source domain model to the target domain.

The most popular line of UDA researches focus on domain invariant representation learning (DIRL) [Pan et al., 2010; Ganin et al., 2016]. Given that the input distribution shift exists between domains, these methods attempt to learn similar feature distributions for both source and target domains, thus

---

* Equal contribution.
† Corresponding author.
In this work, we shed new light on cross-domain NER by introducing a subword-level solution, X-Piece, to directly alleviate the input word-level distribution shift. As shown in Figure 1, we re-tokenize the source domain input words in order to find an optimal subword distribution that approaches the target subword distribution. To optimize the discrete subword distribution, we formulate it as an optimal transport problem, which is solved to minimize the KL-divergence of the source and target conditional distribution (P(label|subword)). To estimate the target conditional distribution, we use the entity lexicon to obtain distant annotations on the unlabeled target data. Such annotation is noisy since it might mistake a large number of entities that do not exist in the lexicon. However, our approach is relatively robust to the noise as it depends only on the distribution, instead of directly training on noisy labels. Additionally, as an input-level solution, X-Piece is complementary to DIRL methods that deal with the feature-level distributions. Experimental results show that the proposed method X-Piece obtains consistent gains based on BERT-tagger over different datasets. Also, the proposed method is shown to benefit the DIRL methods such as DANN.\footnote{Our code is available at https://github.com/trmnwoo/X-Piece. More details and the Appendix can be found at the arxiv version.}

To summarize the contribution of this work:
- We present a first attempt to solve the domain adaptation problem via a subword-level distribution alignment.
- We propose X-Piece, a novel method based on optimal transport to re-tokenize the source domain inputs for alleviating distribution shift.
- Experimental results show the effectiveness of X-Piece based on BERT-tagger on four NER datasets.

2 Related Works

Unsupervised Domain Adaptation Unsupervised domain adaptation (UDA) is a widely concerned problem in machine learning, whose target is to train a model that performs well on a target domain test set with only unlabeled data in the target domain and sufficient labeled data in the source domain. Typical UDA researches include: (1) the feature-based methods [Pan et al., 2010; Ganin et al., 2016] which attempts to learn a feature encoder that maps the source and target inputs into similar feature distributions, thus a classifier trained on the source domain can easily generalize to the target domain. However, as these methods manipulate on the high-dimensional feature distribution instead of directly solving the input data distribution shift, they are less suitable for word-level classification tasks that largely relates to word-level distribution. (2) the instance-based methods [Jiang and Zhai, 2007; Cortes et al., 2008] that leverage the weights of the input instances in order to alleviate the input distribution shift. These methods are similar to our method. However, we directly change the input distribution in a subword-level instead of softly weighting, which is more effective.

Domain Adaptation for NER Several researches have studied cross-domain NER [Chen and Moschitti, 2019; Liu et al., 2021b]. These methods mainly focuses on feature-based [Jia and Zhang, 2020] or data-based [Zhang et al., 2021] domain adaptation, while our approach is completely different from these methods as we are the first to deal with the subword-level distribution for domain adaptation.

Subword Segmentation in NLP Subword segmentation has become a standard process in NLP, which segments input words into subword units to alleviate the problem of infrequent words with a acceptable vocabulary size [Devlin et al., 2019; Liu et al., 2019]. However, typical segmentation approach, such as the WordPiece Tokenizer used in BERT [Devlin et al., 2019], splits words into unique sequences with a greedy longest-match-first algorithm, only considering the frequency and length of subwords for word segmentation. Such approach fails to perform various and task-specific tokenization on downstream tasks. To improve the problem, [Kudo, 2018] and [Prokhorov et al., 2020] propose regularization methods to allow multiple segmentations for a word, while they do not fix the domain shift problem. [Schick and Schütze, 2020; Liu et al., 2021] propose methods to generate embeddings for the rare words and the open-vocabulary words under pretrain-finetune domain shift. Similar to our approach, [Sato et al., 2020] focuses on domain adaptation and adapts the embedding layers of a given model to the target domain. Generally, these DA methods are label-agnostic embedding-level methods trying to alleviate the problem of different marginal distribution $P_S(x) \neq P_T(x)$ between domains. While our method aims at aligning the conditional or joint distribution $(P_S(y|x) = P_T(y|x) \text{ or } P_S(x,y) = P_T(x,y))$ under domain shift from a subword segmentation perspective.

3 Approach

In this section, we introduce a subword-level domain adaptation approach, X-Piece, for cross-domain NER.

3.1 Problem Statement

In this work, we study cross-domain NER following the unsupervised domain adaptation setting. Specifically, we assume a source domain $D_S$ and a target domain $D_T$ with different distribution over $X \times Y$. We are provided with a labeled dataset drawn from the source domain $S = \{X^i, Y^i\}_{i=1}^m \sim D_S$ and an unlabeled dataset drawn from the target domain $T = \{X^i, Y^i\}_{i=1}^m \sim D_T$, where $X^i = \{x_1^i, \ldots, x_n^i\}$ and $Y^i = \{y_1^i, \ldots, y_m^i\}$ are the input word sequence and corresponding labels of the $i$th input sample. DA assumes that the two domains have different marginal distribution $P_S(x) \neq P_T(x)$ and different conditional distribution $P_S(y|x) \neq P_T(y|x)$, thus have different joint distribution $P_S(x,y) \neq P_T(x,y)$. Then, the aim of cross-domain NER is to train a classifier $f : X \rightarrow Y$ using $S$ and $T$ that performs well on the target domain test set $T_{test}$.

3.2 Domain Adaptation via Optimal Word Tokenization

To alleviate the domain shift in the input-level, we propose to align the subword distribution in the source domain towards the target distribution via re-tokenizing the input words. We formulate and solve this discrete optimization problem as an optimal transport problem. An overview of the proposed method is shown in Fig.2.
**KL-divergence as Optimization Objective**

Formally, we assume an identical subword space $T$ and a default tokenizer $T_{ori}$ in both domains. Given any input word $x$, the default tokenizer tokenizes $x$ into a subword sequence $T_{ori}(x) = \{t_1, \ldots, t_k\}, t_i \in T$. Typical tokenizers used in pre-trained LM, such as the Wordpiece tokenizer in BERT-tagger, is usually an injection function, meaning that an input word $x$ is always tokenized into one identical subword sequence regardless of the context and the label. Therefore, we also have $P_{S,ori}(t, y) \neq P_{T,ori}(t, y)$ and $P_{S,ori}(y|t) \neq P_{T,ori}(y|t)$.

Consequently, to solve the distribution shift problem across domains, we are to search for an optimal tokenizer $T^*$ in the source domain such that $P_{S}(t, w, y) \approx P_{T,ori}(t, y)$ (joint) or $P_{S}(y|t) \approx P_{T,ori}(y|t)$ (conditional). Here, we calculate the KL-divergence score as the measurement of distribution distance. The problem can then be formulated as:

$$T^* = \arg \min_T D_{KL}(P_S|P_T)$$  

(1)

where $P$ denotes the joint or conditional distribution.

**Formulating Optimal Transport (OT) Problem**

When searching for the optimal tokenizer $T^*$, we are actually searching for an optimal joint distribution $P_S(t, w, y)$ of subwords, words, and labels, and this joint distribution is to guide the tokenization of each word under a certain label. As a result, this tokenization process can be regarded as transporting the subword distribution $P_S(t)$ into the word-label distribution $P_S(w, y)$. Therefore, we can formulate it as a **discrete optimal transport problem** with a cost considering the source-target discrepancy, which can then be solved efficiently via the Sinkhorn algorithm.

Formally, we reformulate the KL-divergence objective as the following objective function which has the same form as the objective function in optimal transport (The detailed derivation can be found in Appendix):

$$\min_{P \in \mathcal{P}(l^0 \times l)} < P, D > - \gamma \mathcal{H}(P)$$

(2)

Here, $P$ is exactly the optimal joint distribution $P_S(t, w, y)$ we search for, which is then to guide the optimal tokenizer $T^*$. $D$ is the transporting cost matrix related to the distribution distance between domains, which has different form when optimizing based on the joint and conditional distribution:

$$D(t, w, y) = \begin{cases} - \log \left( \frac{P_T(t, y)}{P_T(t)} \right), & \text{Conditional} \\ - \log \left( \frac{P_T(t, y)}{P_T(t)} \right), & \text{Joint} \end{cases}$$

(3)

**3.3 Setup of Optimal Transport**

In the above formulated OT problem, we are to find the best transporting mass from the subword distribution to the word-label distribution. Here we further ensure the validation of the transport solutions by adding some constraints.

First, we set the initial values of each $(w, y)$ term and each $t$ term based on the word-label distribution in the source domain:

$$A(w, y) = \phi(w, y) \cdot |\text{sub}(w)|$$

(4)

$$A(t) = \sum_w \phi(w) \cdot |\text{I}[t \in \text{sub}(w)]|$$

where $\phi(w, y)$ and $\phi(w)$ denotes the frequency of the word-label pair $(w, y)$ and the frequency of the word $w$ itself in the source data. $\text{sub}(\cdot)$ is the set of all the possible subwords segmented from the word $w$. $\text{I}[t \in \text{sub}(w)]$ is a indicator function that stays true if the subword $t$ exists in $\text{sub}(w)$.

Next, considering that the word-label distribution and the possible subwords that can be tokenized from a certain word is fixed, we add the constraints to keep the distributions of each row and column:

$$\sum_{(w, y)} P(t, w, y) = A(t), \sum_t P(t, w, y) = A(w, y)$$

(5)

Then, we fix the cost matrix $D$ considering that a subword should not transport to a word that does not cover it:

$$D(t, w, y) = \begin{cases} D(t, w, y), & \text{if } t \in \text{sub}(w) \\ +\infty, & \text{if } t \notin \text{sub}(w) \end{cases}$$

(6)
## 4 Experiments

In this work, we conduct experiments on cross-domain NER following the commonly adopted unsupervised domain adaptation (UDA) setting. Specifically, we assume an identical label space of both source and target domains. We use the source domain labeled data and the target domain unlabeled data for model training, and then compare the performance on the target domain test set.

We conduct experiments on four commonly used NER datasets from different domains: CoNLL’03 from the newswire domain, Twitter from the social media domain, Webpages from the web domain, and OntoNotes 5.0 which contains text documents from six domains, including broadcast conversation (BC), broadcast news (BN), magazine (MZ), newswire (NW), web (WB) and telephone conversation (TC). The training set size of these sub-domains are 10.4k, 9.7k, 6.9k, 15.2k, 11.1k, 6.4k, respectively. More details of the datasets are shown in Table 1.

To fully illustrate the proposed method, we adopt two experimental settings: (1) Cross-domain between the OntoNotes sub-domains. (2) Cross-domain from CoNLL’03 to other datasets. To ensure the same label space, when conducting experiments on Setting (2), we consider only PER, LOC, ORG entity classes (For OntoNotes, there are several fine-grained entity types such as NORP that can be included by LOC, thus we merge these entity types as LOC). When conducting experiments on Setting (1), we consider all the 18 entity classes.

### 4.1 Baselines

We include several competitive baselines to verify the effectiveness of the proposed method:

**BERT-tagger** [Devlin et al., 2019] The BERT-based baseline for NER which replaces the LM head of pre-trained BERT model with a token classification head. We use the implementation of Huggingface\(^3\) based on the bert-base-cased pre-trained model (also for all the other baselines).

**DANN** [Ganin et al., 2016] and **CMD** [Zellinger et al., 2017] are typical DIRL methods for UDA. DANN leverages a domain discriminator to adversarially train the feature representation. CMD is a typical metric-based DIRL method which minimize a central moment discrepancy to measure the domain distribution shift. Both methods are implemented based on bert-base-cased model.

**EADA** [Zou et al., 2021] is a recently proposed DIRL method for cross-domain text classification task, which leverages an energy-based autoencoder to adversarially train the feature extractor.

**Multi-cell LSTM** [Jia and Zhang, 2020] A competitive cross-domain NER method that trains the source model in a multi-task manner with a language modeling task, an entity prediction task and an attention scoring task, which also requires lexicon-annotation. We implement this method using the same lexicon as ours.

**X-Pieces** The proposed method is implemented based on the bert-base-cased pre-trained model and default tokenizer. The lexicon we used is from [Liang et al., 2020]. For all experiments except that in Section 4.6, we implemented based on

### 3.4 Re-tokenizing Input Words with OT Solution

After obtaining a solution of the distribution \(P(t, w, y)\) via OT, we are to re-tokenize the source domain input words based on \(P(t, w, y)\). However, given the subword distribution \(P(t|w, y)\) of each word-label pair, there is still a gap between the segmented subword sequence and the distribution. Here, we denote a certain segmentation of word \(w\) as \(s = \{t_1, \ldots, t_k\}\). Calculating the distribution of each segmentation \(P(s|w, y)\) based on \(P(t|w, y)\) can be regarded as solving a system of linear equations, where for each \(t_i \in \text{sub}(w)\):

\[
P(s_1) \cdot C_{s_1, t_1} + \cdots + P(s_k) \cdot C_{s_k, t_1} = P(t_1|w, y)
\]

where \(C_{s_k, t_i}\) denotes the number of subword \(t_i\) that contained by the segmentation \(s_k\).

As the coefficient matrix of these linear equations are sparse, we simply assume that each segmentation \(s_j\) include a singular subword \(t_i\) that does not appear in other segmentation, and take the value \(P(t_i)\) as \(P(s_j)\) by:

\[
P(s_j) = \min_{t_i \in s_j} \{P(t_i|w, y), t_i \in s_j\}
\]

Then, given a word \(w\) with label \(y\) in the source domain, we re-tokenize it by sampling a segmentation based on \(P(s|w, y)\) instead of consistently choosing the highest one, in order to match the \(P(t, w, y)\) distribution we have solved:

\[
T^*(w|y) = \text{Sample}(s) \sim P(s|w, y)
\]

| Datasets     | Domain        | # Class | # Train | # Test |
|--------------|---------------|---------|---------|--------|
| CoNLL’03     | News          | 3       | 14.0k   | 3.5k   |
| OntoNotes    | General       | 18      | 60.0k   | 8.3k   |
| Twitter      | Social Media  | 3       | 2.3k    | 3.8k   |
| Webpages     | Website       | 3       | 385     | 135    |

Table 1: Dataset details. Here we only show the actual class number we use in cross-domain settings.
4.2 Cross-domain Performance

Table 2 and Table 3 show the cross-domain results on setting (1) and (2), respectively. From the tables, we can observe that: 1) The proposed X-Piece method consistently exceeds BERT-tagger in most scenarios. In OntoNotes cross-domain setting, X-Piece achieves up to 4.58% improvement over BERT-tagger when transferring from TC to MZ. As TC and MZ are two domains of the smallest training set (6.4k, 6.9k), we induce that X-Piece can bring more advantages on lower-resource cross-domain setting, where the data distribution shift causes more impact on the domain generalibility when trained with less data. 2) The feature-based methods (CMD, EADA) do not always benefit the performance when adapted to NER task, which verifies that transferring the high-dimensional features might not always be effective in word-level classification tasks like NER. 3) X-Piece can also benefit the performance when combined with the feature-based method (DANN), showing the effectiveness of the proposed method as a novel sub-word level domain adaptation tool.

4.3 Correlation with KL-divergence

As described in Section 3.2, we use the KL-divergence score as a measure of the domain distribution shift, and optimize the subword distribution to minimize the KL score. How does the KL divergence vary after OT and how is the variation related to the performance? In Fig.3, we show the performance gain and the deviation of the KL-divergence before and after OT (WB to other OntoNotes sub-domains). Intuitively, we can observe that the KL-divergence between the source subword distribution and the target lexicon-approximated distribution consistently declines after optimized by OT. Also, the $|\Delta(KL)|$ shows a positive correlation with the performance gain, which verifies the rationality of using KL-divergence as the domain shift measure.

4.4 Impact of Lexicon Annotation

As mentioned in Section 3.3, we use the lexicon annotation on the target domain unlabeled data for estimating the target distribution. In this section, we conduct experiments (WB to other OntoNotes sub-domains) to investigate the impact of lexicon noise, as shown in Tab.4. We train the BERT-tagger directly on the lexicon annotated data under two settings. The further setting fine-tunes BERT-tagger on the source data and then further fine-tunes on the target lexicon annotated data; The together setting fine-tunes the model on the source data and target lexicon annotated data simultaneously. We can observe that the lexicon-annotated data can hardly benefits the model performance or even causes degradation when trained directly. While the proposed method, which only use the noisy data to estimate the target distribution, is relatively robust to the annotation noise.

To further validate the proposed method, we also calculate the KL-divergence between the lexicon-approximated distribution and the gold-labeled distribution (Lexicon $\parallel$ Gold). As a comparison, we additionally show the KL-divergence between source distribution (Source $\parallel$ Gold), uniform distribution (Uniform $\parallel$ Gold) and gold-labeled distribution. The results are shown in Fig.5. We can observed that the KL-divergence optimizine the conditional distribution. We also combined our method with DANN for further improvement. We implement all models in MindSpore.
between lexicon-approximated distribution and target distribution is lower than others, indicating that using the lexicon to estimate the target distribution is feasible.

### 4.5 Computational Cost

X-Piece contains a pre-processing process to calculate the distribution via OT, which can be done offline on a CPU machine. After one pass of pre-process, the distribution matrix can be stored for future tokenization. In this section, we conduct experiments to investigate the computational cost of the pre-processing part (OT) of X-Piece. Specifically, we test the computation time against different corpus size on Intel Xeon Platinum 8260, 2.40GHz. As shown in Fig.4, the word number and the subword number increase as the corpus size rises, meaning that the distribution matrix to be solved is also expanding. As a result, the computation time of the proposed method also increases. However, the computational cost is acceptable (up to 58.84s) even with a corpus size of 59.9k sentences and a distribution matrix size of 30.8k × 7.8k.

![Figure 4: Computational cost of the pre-processing of X-Piece against corpus size. The computational cost is acceptable (up to 58.84s) even with a distribution matrix size of 30.8k × 7.8k.](image)

### 4.6 Optimizing with Joint Distribution

In Section 3.2, we formulate the optimal transport problem with two optimization objective, the conditional distribution (\(P(y|x)\)) and the joint distribution (\(P(x,y)\)). The difference is that the conditional objective only cares about the relative distribution of labels given a certain subword, while optimizing the joint objective means additionally aligning the subword distributions. In Table 5 we compare these two methods on WB to other OntoNotes sub-domains. Generally, the performance of the conditional objective is better than the joint objective. However, in some cases (WB to MZ, WB to TC) the joint objective show advantages. This might because the training set size of MZ and TC is relatively small, thus optimizing the joint distribution is feasible. In such cases, optimizing the joint distribution can better decrease the domain gap as it additionally solves the shift of \(P(t)\).

![Figure 5: KL-divergence of the lexicon-approximated distribution against gold distribution (Lexicon || Gold), compared to other distributions. These results validate the usage of lexicon for estimating target distribution.](image)

### 5 Conclusion

In this work, we propose X-Piece, a subword-level approach for cross-domain NER, which alleviates the distribution shift between domains directly on the input data via subword-level distribution alignment. Specifically, we re-tokenize the input words in order to align the subword distribution with an lexicon-approximated target distribution. To find the optimal subword distribution, we formulate and solve it as an optimal transport problem. Experimental results verify the effectiveness of the proposed method on four NER datasets from different domains. Also, the proposed method is shown to benefit other cross-domain algorithms.
Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 62076069, 61976056), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). This research was supported by Meituan and CAAI-Huawei MindSpore Open Fund.

References

[Chen and Moschitti, 2019] Lingzhen Chen and Alessandro Moschitti. Transfer learning for sequence labeling using source model and target data. In AAAI, 2019.

[Cortes et al., 2008] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In ALT, pages 38–53, 2008.

[Devlin et al., 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In NAACL, pages 4171–4186, June 2019.

[Du et al., 2021] Xin Liu, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Min Zhang, Haiying Zhang, and Jinsong Su. Bridging subword gaps in pretrain-finetune paradigm for natural language generation. In ACL, pages 6001–6011, August 2021.

[Liu et al., 2021b] Zihan Liu, Yan Xu, Tiezheng Yu, Wenhong Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. AAAI, pages 13452–13460, May 2021.

[Long et al., 2017] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In ICML, volume 70 of Proceedings of Machine Learning Research, pages 2208–2217, 06–11 Aug 2017.

[Pan et al., 2010] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. IEEE transactions on neural networks, 22(2):199–210, 2010.

[Peng et al., 2018] Minlong Peng, Qi Zhang, Yu-gang Jiang, and Xuanjing Huang. Cross-domain sentiment classification with target domain specific information. In ACL, pages 2505–2513, July 2018.

[Provilkov et al., 2020] Ivan Provilkov, Dimitrii Emelienko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In ACL, pages 1882–1892, July 2020.

[Sato et al., 2020] Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. Vocabulary adaptation for domain adaptation in neural machine translation. In Findings of EMNLP, pages 4269–4279, November 2020.

[Schick and Schütze, 2020] Timo Schick and Hinrich Schütze. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In ACL, pages 3996–4007, July 2020.

[Shah et al., 2018] Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. Adversarial domain adaptation for duplicate question detection. In EMNLP, pages 1056–1063, October-November 2018.

[Yosinski et al., 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In NIPS, volume 27, 2014.

[Zellinger et al., 2017] Werner Zellinger, Thomas Grubinger, Edwin Lughhofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. arXiv preprint arXiv:1702.08811, 2017.