Learning Cross-language and Cross-style Mappings with Limited Supervision

Ruochen Xu

CMU-LTI-19-009

July, 2019

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Yiming Yang, Chair (Carnegie Mellon University)
Jaime Carbonell (Carnegie Mellon University)
Graham Neubig (Carnegie Mellon University)
Ming Zhou (Microsoft Research Asia)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Transfer Learning, Semi-supervised Learning, Unsupervised Learning, Deep Learning, Natural Language Processing, Cross-lingual NLP, Text Classification, Word Embedding, Text Generation

To Ruonan, the love of my life, without whom this thesis could have been completed three months earlier.

Abstract

Recent natural language processing (NLP) research has been increasingly focusing on deep learning methods and producing superior results on various NLP tasks. Deep NLP models are usually based on the dense vector representation of input and are able to automatically extract multi-scale features given human-annotated data. However, human annotations are expensive and often not evenly distributed across different languages, domains, genres, and styles. This thesis focuses on multiple aspects of cross-language and cross-style mapping in text, addressing the limitations of existing methods and improving the state-of-the-art results when sufficient amounts of labeled data are not available. By developing both task-oriented transfer learning models (e.g., for class-language classification) and generic methods for mapping among embedded words or sentences, the key contribution of this thesis is a set of novel approaches to leveraging unlabeled text data for effective and efficient mapping across languages or styles.

Chapter 1 outlines the overall theme, challenges being addressed and unique contributions in this thesis.

Chapter 2 presents two novel methods for the transfer of trained text classification models from rich-resource languages to low-resource languages. The first model focuses on the scenario where only a bilingual dictionary of limited size is available as the linkage between languages. It uses unsupervised word embeddings trained on monolingual data to construct a regularization graph in each language and a spectral graph propagation algorithm to extend bilingual dictionaries. The second model is a distillation approach over parallel data in the scenario where the teacher network and student network are classifiers in different languages, respectively. Both models achieved state-of-art performance at the time on several benchmark datasets [60, 62, 84, 129]

Chapter 3 presents an unsupervised approach to the mapping of monolingual word embeddings across languages. It is the first gradient-based method for optimizing the Sinkhorn distance between two spaces of word embeddings and has been proven to be more accurate and robust than other methods [58, 126]. More importantly, this model achieves the state-of-art performance without using any bilingual dictionary or parallel data.

Chapter 4 present new text generation models that transfer the styles or attributes of sentences. We introduce a semi-supervised model which is trained on both paired and unpaired sentences with style labels and achieved the state-of-art results in a formality transfer task. For the case where no paired sentences are available, we proposed a novel unsupervised method that combines the strength of neural Seq2Seq model and search engine, and outperforms other competing methods on various datasets.

Acknowledgments

Throughout the writing of this dissertation, I have received a great deal of support and assistance. First and foremost I would like to thank my advisor Yiming Yang, whose insight, enthusiasm and immense knowledge have been invaluable. I have learned so much from Yiming since my first day at CMU and the experience was truly wonderful. I could not have imagined having a better advisor for my Ph.D. study.

In addition to Yiming, I would like to express my gratitude to the rest of my thesis committee, Jaime Carbonell, Graham Neubig and Ming Zhou for their insightful feedback and guidance on my work. I really appreciated all their assistance.

I am grateful to all of my collaborators, research group members, and friends in Pittsburgh, including Eduard Hovy, Teruko Mitamura, Lori Levin, Alan W Black, Florian Metze, Patrick Littell, David R Mortensen, Zaid Sheikh, Siddharth Gopal, Hanxiao Liu, Wanli Ma, Guoqing Zheng, Yuexin Wu, Wei-Cheng Chang, Jingzhou Liu, Guokun Lai, Naoki Otani, Bohan Li, Weiran Xu, Zihang Dai, Donghan Yu, Ruohong Zhang, Yong-Siang Shih, Zhengbao Jiang, Hiroaki Hayashi, Tian Tian, Aldrian Obaja Muis, Nidhi Vyas, Jiateng Xie, Zirui Wang, Yuntian Deng, Mingda Zhang, Bishi Wang, Chen Kong, Jinmei Zheng, Weiguang Mao, Jin Hu and Qi Guo.

Many thanks to my family, especially my parents, who always unconditionally love me and encourage me to achieve my best.

Finally, I thank with love to my wife Ruonan, who loved, supported, entertained and helped me through this agonizing and amazing journey. I feel so fortunate to have you in my life.

Contents

List of Figures

List of Tables

Chapter 1

Overview

Given an object in one domain, we defined the mapping problem in NLP to be the task of finding its equivalence in another domain. The objects could be basic language units such as words and sentences, or task-oriented models such as topic classifiers or named entity recognizer. The domain could be language, genres, style and so on. For instance, the mapping problem of words across languages is bilingual lexicon induction. The one of sentences across styles is text style transfer. And mapping task-oriented models across languages is cross-lingual transfer learning. The learning of such mappings usually requires a large quantity of supervised data which are in the form of paired words or sentences between the domains. However, supervision is limited or even not exist for many domains of interest. The general goal of this thesis is to enable and enhance the cross-language and cross-style mappings under such low-resource settings. To achieve this, we proposed a set of novel methods for three representative mapping problems.

The first problem we addressed is the problem of cross-lingual text classification (CLTC) in chapter 2, where training data and testing data are in different languages. We started from this specific problem because 1) language is a very important and common domain of text 2) classification is a basic and well-studied application and 3) CLTC under low-resource setting was not well-studied in existing works. The second problem is cross-lingual word embedding (CLWE) in chapter 3. The task is to map words into unified dense vector space where words in different languages are close to each other if they are translations to each other. Comparing with the CLTC, CLWE is natural generalization because word embedding provides the input for many neural NLP tasks including but not limited to classification. Since word translation could be easily induced via nearest-neighbor search in a unified vector space, we could view CLWE as a word-level transfer of textual data. After investigating model-level and word-level mapping learning under limited supervision, we further investigate sentence-level mapping learning with respect to the problem of writing style and attribute transfer (SAT) in chapter 4. The task is to re-write a sentence in a different style or change one of its attributes while keeping the other orthogonal content unchanged. The motivation for studying SAT under limited or zero supervision is the prohibiting cost of producing the parallel data. And the reason for which we switch the domain of mapping from language to style is that low-resource MT has already developed some promising solutions such as pivoting and multi-task learning. Those techniques, however, are not directly applicable in the problem of SAT.

Table 1.1 summarizes the different tasks, domains and granularity levels of the mapping

learning problems we focus on in this thesis.

Chapter	Task	Domain of Interest	Mapping Level
2	Text Classification	Language	Classification Model
3	Word Embedding	Language	Word/Model
4	Text Generation	Style/Attribute	Sentence

Table 1.1: Summarize of Chapter 2 to 4 about the corresponding task, domain and mapping level.

In spite of the difference in tasks, domains, and levels of mapping, the approaches we proposed all share the common requirement of only limited or even zero supervision. To achieve this, our models improved upon the previous works by defining more sophisticated objectives on the unpaired data, such as words that belong to a certain language but not in the bilingual dictionary or the sentences that belong to a certain style/attribute but not in the parallel corpora.

For CLTC, we studied the transfer of classification models. In other words, a classification model is firstly trained in the source language with standard supervised technique, and then is transferred to the target language of interest. The latter step requires some correspondence data between the source and target languages (and domains), such as bilingual lexicons and parallel sentences. We first developed a novel model [117] that transfers the classifier with a limited amount of bilingual dictionary. The incomplete bilingual dictionary is extended via a transductive label propagation algorithm, and then the extended dictionaries are used to transfer the classification model. Compared with existing methods that are only based on the incomplete bilingual dictionary, our method also incorporates the unlabeled words and their monolingual similarities. The second model [116] we introduced is a generalized distillation model under the cross-lingual setting where the supervision is in the format of parallel sentences. Our model is unique in following two ways: 1) it learns task-specific representations instead of general ones by transfer knowledge in the classification label space 2) it addresses the domain mismatch between the parallel corpus and the classification data with a novel adversarial feature adaptation.

For CLWE, we developed a new approach [118] to establish word-level mapping across languages. The key idea is to minimize the distributional distance between the transferred embedding of source-language words and the embedding of target-language words. Specifically, by using the Sinkhorn distance as the distributional distance and with proper constraints on the transformation function, our method enables more robust and effective unsupervised learning of the cross-language alignment across the two embedding spaces and obtained the state-of-the-art results on extensive language pairs.

For SAT, we first explored in the semi-supervised setting where a small quantity of parallel data plus a large quantity of unpaired sentences is available. Our omnivorous model takes data of both types and effectively combines the advances in unsupervised SAT and semi-supervised MT, achieving a new state-of-the-art on a formality transfer dataset. Our second approach focuses on the pure unsupervised setting, where only unpaired sentences are available. We addressed the limitations of previous works, namely the difficulty to disentangle attribute from

content and the error propagation of keyword removal, with a novel model architecture that combines a sequence-to-sequence model and a search engine.

Chapter 2

Cross-lingual Transfer of Text Classification Models

2.1 Motivation

The massive amount of multilingual documents on the World Wide Web makes the cross-lingual text categorization (CLTC) problem increasingly important, whose solutions aim to provide organizational views of the data. Typically, CLTC refers to the task of classifying documents in different languages using the same taxonomy of predefined categories. The Reuters News Agency, for example, has been using the same taxonomy of subject topics to index International news stories in different languages. Automated classification of multilingual documents is desirable for both cost-saving and classification consistency.

The CLTC problem would be relatively easy to solve if we had a sufficient amount of labeled training data for each language because most machine learning techniques for text classification have the flexibility to be applied to any language. However, for many languages in the real world, a large quantity of human-labeled documents for training classifiers is often hard to obtain. Thus a natural solution is to train classifiers in a label-rich language and then apply the trained classifiers to documents in label-poor languages. For convenience let us denote the language that provides labeled documents for training classifiers as the source language, and the other languages that provide unlabeled test documents as the target languages. How to successfully apply the trained classifiers in the source language to documents in different target languages is the key question for research.

Existing CLTC methods differ in how to make the classification across languages. Bel et al. [10] presented an early effort where they translated the target-language documents to the source language using a comprehensive bilingual dictionary, and then applied the classifiers in the source language to the translated documents. To reduce the computational cost, they only translated the topically important terms in those documents. Similarly, Ling et al. [64] also translated target-language documents (Chinese web pages) to a source language (English), and predicted their labels based on the labels of the English documents which are similar to the translated versions of the Chinese documents.

Rigutini et al. [91] translated training documents from a source language to a target lan-

guage instead, and applied an Expectation Maximization (EM) algorithm to leverage unlabeled documents in the target language in addition. In the E-step the unknown labels were guessed for the target-language documents, and in the M-step the classifier parameters were updated based on both the (translated) training documents with true labels and the target-language documents with guessed labels.

Wan [111] used machine translation (MT) systems to perform English-to-Chinese and Chinese-to-English translation of each document in a collection of labeled English and unlabeled Chinese documents. The original document (before translation) and its translated version were called the *two views* of the same documents. The two views of all the documents enabled a co-training algorithm to train and re-train classifiers both in English and Chinese alternately and iteratively. That is, it started with the labeled portion of the documents as the initial training set, and then added more and more classifier-assigned labels to the unlabeled portion of the documents for retraining. This process resulted in improved classifiers in both languages while human-labeled documents were available only in one language. With the help of machine translation, some recent works also solved CLTC via multi-view learning methods, including majority voting[2], multi-view co-regularization[38] and representation learning[37].

Instead of translating documents as in the above approaches, Shi et al. [98] tried to translate classification models across languages. The source-language model of each category consisted of a bag of weighted terms, where the term weights were the learned model parameters based on labeled data. Then each term in the model was translated to the target language based on a comprehensive bilingual dictionary. To handle ambiguities (one-to-many mapping) in term translation, an EM algorithm was used to obtain the cross-lingual translation probabilities.

2.2 Transfer via limited cross-lingual dictionary (CIKM' 16)

While the relevant literature has provided valuable insights about how to tackle the CLTC problem, existing methods have an implicit or explicit assumption in common, i.e., the availability of rich cross-lingual knowledge resources for each language pair of interest. By rich knowledge resources here we mean comprehensive bilingual dictionaries and MT systems for quality-translation of documents or classification models in the domains of interest. Such an assumption would significantly limit the generalization or applicability of those methods to a broad range of low-resource languages. In fact, except the dominating or most common languages (like English, French, Spanish, etc.), the majority of languages in the real world often do not have large quantities of comprehensive cross-lingual dictionaries or high-quality MT systems to support CLTC in every possible domain of interest. This fact makes the CLTC challenge wide open, i.e., we must solve the problem without relying on the availability of rich cross-lingual knowledge resources. How do we get there? Existing research in CLTC has not answered this question.

This section focuses on the open challenge of low-resource CLTC, especially under the condition where the bilingual dictionaries are highly incomplete and very small in size. We further narrow down our focus on translating classification models across languages instead of translating documents, as the former is computationally much more efficient than the latter

(when the document collections are very large), and often the solutions of the former can be easily generalized to the latter in principle.

We propose a novel label propagation algorithm to extend the incomplete bilingual dictionaries. Given the extended bilingual dictionary, we further introduce a simple yet effective model translation method to translate the source-language classification model to the target language. More specifically, our proposed CLTC model for the low-resource setting has the following two steps:

- Firstly, we take monolingual corpus and induce monolingual word similarities in both the source and the target language. The prediction for the missing bilingual dictionaries are optimized jointly based on the observed dictionary entries across the languages and the monolingual word similarities within both the source and the target language.
- Secondly, a Naive Bayes model trained in the source language is translated to the target language based on the predicted bilingual dictionaries from the first step.

2.2.1 Cross-lingual Dictionary Extension

We explored two sets of approaches to the statistical extension of bilingual dictionaries for CLTC, conditioned on the availability of a small-sized (incomplete) dictionary per language pair. Both categories of methods combine strengths of unsupervised word embeddings in each language and supervised or semi-supervised mapping of words across languages. The two categories differ in how to establish the mapping. The first set of methods are inspired by the research on cross-lingual word embedding [6, 72, 100] and uses supervised learning to obtain a set of linear models for cross-lingual word mapping, where the true translation pairs of words in the initial dictionary are treated as labeled training pairs.

The second method is a novel transductive learning method that jointly leverages both labeled and unlabeled word pairs across two languages in the optimization of the mapping. We call it the *Transductive Label Propagation (TransLP)* approach. To the best of our knowledge, TransLP is the first transductive method proposed for the problem of bilingual dictionary extension. Comparing with the supervised methods, it also leverages the unlabeled words which are not included in the observed bilingual dictionary. The advantage of TransLP could be verified on the better performance of dictionary extension and CLTC tasks. In addition to the advanced methods we propose, we also include a simple and intuitive baseline for comparison, which we call the *k Nearest Neighbor (kNN)* method.

Supervised Cross-lingual Word Embedding

1) Regularized Linear Regression (RidgeReg): Word embedding has been a hot topic in recent machine learning and was found effective in capturing semantic similarities among words when trained on large document collections. It discovers a vector representation of each word based on its co-occurrence patterns with other words, and enables inference about some semantic relations via simple operations. For example, *vector(England) - vector(London)* would be similar to *vector(China) - vector(Beijing)* [74]. More interestingly, Mikolov et al. [72] showed that such linear relations exist in different languages, and the relations can be easily translated across languages using multivariate linear regression.

Intrigued by this line of work we propose to extend an existing (small-sized) bilingual dictionary via cross-lingual regression over embedded words. Denote by $\mathcal{D} = \{(x_i, z_i)\}_{i=1}^{n}$ the given bilingual dictionary, where $x_i \in \mathbb{R}^{d_1}$ is the vector representation of word i in the source language and $z_i \in \mathbb{R}^{d_2}$ is the vector representation of word j in the target language with equivalent meaning. Using the word pairs in the dictionary as the labeled training set, we can learn a transformation matrix with the following objective function:

$$\min_{W \in \mathbb{R}^{d_2 \times d_1}} \sum_{i=1}^{n} \|Wx_i - z_i\|^2 + \lambda \|W\|_F^2 \tag{2.1}$$

Here W is the unknown matrix we want to optimize; the first term in the function is the training-set loss, and the second is the regularization term, to avoid overfitting on the training data. Notice that optimizing W row-by-row is equivalent to solving a series of ridge regression problems.

The problem in (2.1) has a closed-form solution given by

$$W^* = ZX^T(XX^T + \lambda I)^{-1} \tag{2.2}$$

where $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d_1 \times n}$ and $Z = [z_1, z_2, ..., z_n] \in \mathbb{R}^{d_2 \times n}$.

Once W is learned based on the training set, we can use it to map any x which may be unseen in $\mathcal{D}$ to vector $z = Wx$ in the target space. Then we can find the target-language words as the translations of word x if those words are among the k nearest neighbors of vector z. We treat k as a hyper-parameter of this method, which could be tuned via cross-validation. We use the cosine similarity as the nearness measure among vectors.

2) Orthogonal Regression (OrthReg): Recent works [6, 100] showed that further constraining the W in the linear regression of RidgeReg to be orthogonal would empirically lead to a better mapping between two vector spaces. We followed [6] and solve the optimal W under the constraint as

$$W^* = VU^T \tag{2.3}$$

where $Z^TX = U\Sigma V$ is the SVD factorization of Z^TX.

3) Canonical Correlation Analysis (CCA): We follow the work of Faruqui and Dyer[24] as comparison for our proposed methods with state-of-art approach of cross-lingual word embeddings. They used canonical correlation analysis (CCA) for incorporating multilingual evidence into vectors generated monolingually. Given mapped monolingual word vectors X and Z as defined in (2.2), CCA first seeks v and w such that:

$$v, u = \text{argmax}_{v \in \mathbb{R}^{d_1}, u \in \mathbb{R}^{d_2}} \text{corr}(v^TX, w^TZ) \tag{2.4}$$

v^TX, w^TZ are called the first canonical variate pair. We further seeks vectors maximizing the same correlation but subject to the constraint that they are to be uncorrelated with the first canonical variate pair. This process may continue to d times, where $d = \min(d_1, d_2)$. The resulting matrix $V \in \mathbb{R}^{d_1 \times d}$ and $W \in \mathbb{R}^{d_2 \times d}$ are used to map any source word vector x and target word vector z, which may not appear in the bilingual dictionary, to a unified space: $x^\star = V^Tx, z^\star = W^Tz$. We apply the same procedure to find translations in the unified vector space as described in RidgeReg.

Transductive Label Propagation (TransLP)

Although it is a natural choice to establish a linear mapping of embedded words across languages, it does not explore the power of non-linear transformation. Also, it only leverages the labeled data (the true translation pairs of words in the given dictionary), but not the vastly available unlabeled words in both the source language and the target language. To broaden the scope of our investigation, we propose a transductive label propagation (TransLP) approach for a non-linear cross-lingual mapping and for utilizing both labeled and unlabeled data during training.

TransLP is a semi-supervised learning framework that has been developed recently for bipartite link prediction based on multi-source relations [66]. The key idea is to use the graph product operations (such as the Kronecker product) to combine relational information in multi-source graphs, and then to propagate the label information in the observed (labeled) links to the unknown (unlabeled) links over the product graph. Adapting this idea to the cross-lingual mapping of embedded words, we want to propagate the labels of the known translation pairs of words (in the provided dictionary) to the unknown pairs based on word-word similarities within both the source language and the target language.

Denote by $G \in \mathbb{R}^{|V|\times|V|}$ and by $H \in \mathbb{R}^{|V'|\times|V'|}$ the word similarity graphs within the source and target languages, respectively, where $G_{ii'}$ encodes the similarity between word i and word i' in the source language, and $H_{jj'}$ encodes the similarity between word j and word j' in the target language. Specifically, we construct these graphs by computing the pairwise cosine similarities for the embedded words within each language, and by linking two words if and only if they are among the k-nearest neighbors of each other. We further define the induced similarity between the cross-lingual links (i, j) and (i', j') as $G_{ii'}H_{jj'}$, which means that the two links should have similar labels (as word translation pairs or not) if the embeddings of words i and i' are similar in the source language and if embeddings of words j and j' are similar in the target language. An illustration of TransLP with some toy data is shown in figure 2.1

Denote by F_{ij} the system-predicted score for cross-lingual link (i, j). Intuitions above can be encode in the following Gaussian random field prior over $F \in \mathbb{R}^{|V|\times|V'|}$

$$\text{vec}(F) \sim \mathcal{N}(0, G \otimes H) \tag{2.5}$$

where vec is the vectorization operator that concatenates the columns of a matrix into a single vector, $G \otimes H$ stands for the Kronecker product of G and H.

Our optimization objective is defined as

$$\min_{F\in\mathbb{R}^{|V|\times|V'|}} \quad \sum_{i,j} \ell\left(F_{ij}, \mathbf{1}_{(x_i,z_j)\in\mathcal{D}}\right) + \frac{\gamma}{2} r(F) \tag{2.6}$$

where regularization $r(F)$ corresponds to the negative likelihood of the Gaussian random field prior defined by (2.5)

$$r(F) = \text{vec}(F)^\top (G \otimes H)^\dagger \text{vec}(F) \propto -\log p(F \mid G, H) + const \tag{2.7}$$

where † stands for matrix pseudoinverse.

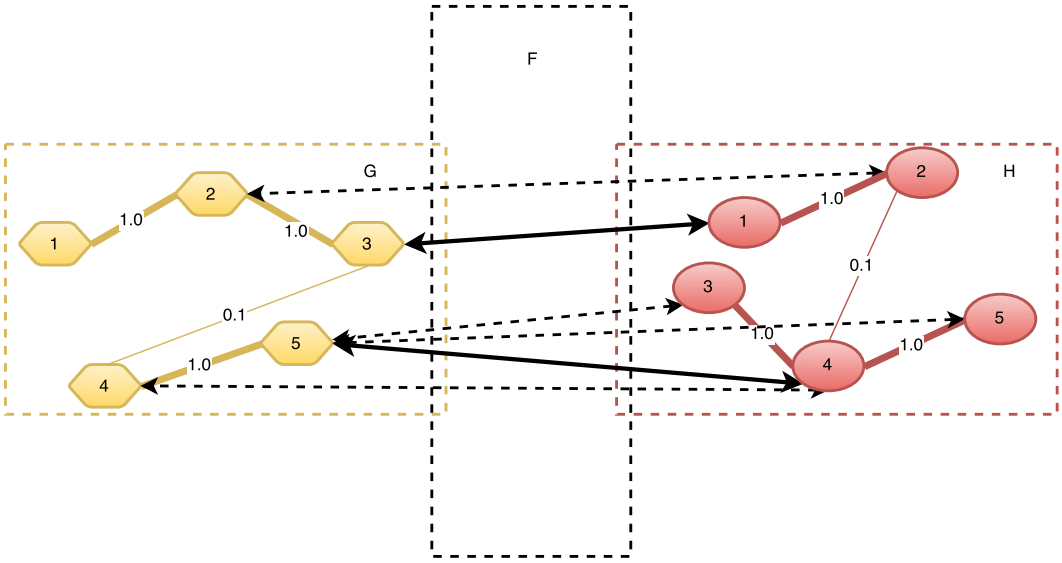


Figure 2.1: Illustration of TransLP with toy examples. Nodes in different color and shape represent words in different languages. G and H contain edges encoded by word-to-word similarity. Solid lines with double arrows are observed bilingual dictionary pairs and dashed lines with double arrows are predicted bilingual pairs.

One may choose any loss function ℓ in (2.6), such as squared error $\ell(x, y) := (x - y)^2$, the indicator function $\mathbf{1}_{\{\cdot\}}$ equals 1 if $(x_i, z_j) \in \mathcal{D}$ and equals zero otherwise. The first term in (2.6) encourages our predictions to fit the observed labels, and the second term encourages the predicted values to have a smooth propagation with respect to the similarities among cross-lingual word pairs.

Optimizing (2.6) would be extremely expensive when $|V|$ and $|V'|$ are large. To speedup, we propose to reduce the computational complexity via low-rank approximations. More specifically, we approximate G and H with their leading eigenvectors $U \in \mathbb{R}^{|V| \times k_1}$ and $V \in \mathbb{R}^{|V'| \times k_2}$, and restrict matrix F within the linear span of those eigenvectors

$$G = \sum_{i=1}^{k_1} \lambda_i v_i v_i^\top \tag{2.8}$$

$$H = \sum_{j=1}^{k_2} \mu_j u_j u_j^\top \tag{2.9}$$

$$F = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij} v_i u_j^\top \tag{2.10}$$

For sparse graphs, both U and V can be obtained via power iteration algorithm in linear com-

plexity over $|V|$ and $|V'|$. The regularization term in (2.6) can be simplified as

$$
\begin{aligned}
r(F) =& \text{vec}(F)^\top (G \otimes H)^\dagger \text{vec}(F) && (2.11)\\
=& \text{vec}(F)^\top (G^\dagger \otimes H^\dagger) \text{vec}(F) && (2.12)\\
=& \langle F, G^\dagger F H^\dagger \rangle && (2.13)\\
=& \Big\langle F, \sum_{i=1}^{k_1} \lambda_i^\dagger v_i v_i^\top \sum_{i'=1}^{k_1} \sum_{j'=1}^{k_2} \alpha_{i'j'} v_{i'} u_{j'}^\top \sum_{j=1}^{k_2} \mu_j^\dagger u_j u_j^\top \Big\rangle && (2.14)\\
=& \Big\langle \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij} v_i u_j^\top, \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij} \lambda_i^\dagger \mu_j^\dagger v_i u_j^\top \Big\rangle && (2.15)\\
=& \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij}^2 \lambda_i^\dagger \mu_j^\dagger && (2.16)
\end{aligned}
$$

We then solve the following optimization problem with a substantially reduced number of model parameters

$$
\begin{aligned}
\min_{\{\alpha_{ij}\}_{i=1,j=1}^{k_1,k_2}} \quad & \sum_{i,j} \ell \left[\Big(\sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij} v_i u_j^\top \Big)_{ij}, \mathbf{1}_{(x_i,z_j)\in D} \right] \\
& + \frac{\gamma}{2} \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_{ij}^2 \lambda_i^\dagger \mu_j^\dagger
\end{aligned} \tag{2.17}
$$

It is not hard to verify that optimization problem (2.17) is convex over the α_{ij}'s. Since typically $k_1, k_2 \ll \min\{|V|, |V'|\}$, each gradient update for the above optimization only takes $O(|V| + |V'|)$ flops. We empirically find it sufficient to set $k_1 \leq 500, k_2 \leq 500$ for good performance in practice.

K-Nearest Neighbor (kNN)

As an intuitive and simple baseline for comparison, the kNN approach for bilingual dictionary extension is defined as the following. For each word pair (w', w) in the initial bilingual dictionary where w' is a word in the source language and w is a true translation of w' in the target language, we extend the dictionary by adding the k words most similar to w in the target language as the valid translations of word w'. Symmetrically, we do such kNN extension for word w' in the source language as well. Notice that each word has a vector representation obtained by applying word embedding to each language, and that the similarity between each word pair in the language is measured by the cosine of the corresponding vectors.

2.2.2 Classification Model Translation

Cross-lingual Naive Bayes (CLNB)

Given a bilingual dictionary (extended using the methods in the above section), we want to translate the classification models trained on the labeled documents in the source language

to the target language. We use a standard multinomial Naïve Bayes (NB) as the classification method [68] because the probabilistic model parameters allow easy translation of NB models in a probabilistic manner[1].

Given document d in the target language, the conditional probability of d being generated from category c is given by:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{2.18}$$

Since $P(d)$ is independent of c, the denominator of (2.18) can be ignored when predicting category label $\hat{y}$ for document d as

$$\hat{y} = \arg\max_{c \in C} P(c)P(d|c) \tag{2.19}$$

where C is the candidate set of category labels, and $P(d|c)$ is the probability of document d conditioned on category c. The latter is proportional to the product of word probabilities under the independence assumption:

$$P(d|c) \propto \prod_{w \in d} P(w|c) \tag{2.20}$$

For engaging with cross-lingual translation probabilities, we specify word probability $P(w|c)$ to be decomposed as:

$$P(w|c) = \sum_{w'} P(w', w|c) = \sum_{w'} P(w'|c)P(w|w', c) \tag{2.21}$$

where word w' is any word in the source language, $P(w'|c)$ is the probability of source word conditioned on category c, and $P(w|w', c)$ is the translation probability form w' to w conditioned on category c.

Given a training set of labeled documents in the source language with the vocabulary size of V', conditional probability $P(w'|c)$ is typically estimated with Laplace smoothing as:

$$\hat{P}(w'|c) = \frac{T_{c,w'} + 1}{(\sum_{v' \in V'} T_{c,v'}) + |V'|} \tag{2.22}$$

where $T_{c,w'}$ be the number of occurrences of word w' in the training documents from category c. As for category prior, we just use the Maximum Likelihood Estimate (MLE) of $\hat{P}(c) = \frac{N_c}{N}$ where N is the size of the labeled training set and N_c is the number of labeled documents in category c. We assume the category priors are the same in both the source and the target languages.

Now the missing part we need for completing formula 2.21 is $P(w|w', c)$. Denoting by $\mathcal{D}$ the given bilingual dictionary, we set $P(w|w', c) = 0$ if pair $(w', w) \notin \mathcal{D}$; otherwise, we estimate it using cross-lingual word similarities with normalization as:

[1]We have also examined other types of classifiers including Support Vector Machines (SVM), and found that the associated model translation is either more complicated or less effective, or both. Details on this are beyond the scope of this paper.

$$P(w|w',c) \approx P(w|w') \approx \frac{\text{sim}(w,w')}{\sum_{v \in D(w')\text{sim}(v,w')}} \tag{2.23}$$

where $D(w')$ is the set of target-language words as the translations of source word w' in the dictionary; $\text{sim}(w,w')$ is the similarity score given by dictionary extension methods for target word w and source word w'. For example, in RidgeReg we have

$$\text{sim}(w,w') = \cos(Wx', x)$$

where x' and x are vector representation for w' and w. Similarly, for CCA we have

$$\text{sim}(w,w') = \cos(V^T x', W^T x)$$

In TransLP we have

$$\text{sim}(w,w') = F_{ij}$$

where i and j are indices for w and w'. And in kNN we have

$$\text{sim}(w,w') = \begin{cases} 1, & \text{if } (w,w') \in \mathcal{D} \\ 0, & \text{otherwise} \end{cases}$$

Our model translation method is computationally very efficient. At testing time, $P(w|c)$ is first computed according to equation (2.21) and stored. Then the classifier makes predictions following equations (2.19) and (2.20) in the same way as in standard monolingual Naïve Bayes classifiers. In practice, it takes less than a second to run our CLTC method over a test collection of a few thousand documents.

A potential weakness of our method, on the other hand, is in its approximation of $P(w|w',c) \approx P(w|w')$. That is, the word embedding components we used and the cross-lingual mapping of embedded words are not category-sensitive. We leave the category-sensitive enhancement of our approach to future research.

Baselines

1) Cross-Lingual Mixture Model(CLMM): The method by Shi et al.[98] also estimates $P(w|w',c)$ to translate their classification model from source language to target language. They exploited the readily available unlabeled data in the target language via semi-supervised learning. To summarize, they optimize $\theta = P(w|w',c)$ to maximize the following log-likelihood:

$$l(\theta) = \sum_{d \in D_u} log \sum_c P(c) \sum_{d' \in D(d)} \prod_{w' \in d'} P(w|w',c) P(w'|c) \tag{2.24}$$

where $P(w'|c)$ and $P(c)$ are learned from training data in source language and viewed as fix parameters; $d' \in D(d)$ represents all possible document d' translated from d according to dictionary $\mathcal{D}$. Note that the model assumes the availability of certain amount of unlabeled documents in target language (i.e. D_u). Those documents are further required to belong to the same taxonomy of data in source language. Those assumptions may not hold in the low-resource scenario.

On the other hand, our proposed model is capable of utilizing more accessible general-purpose monolingual corpus(e.g. Wikipedia) to propagate existing bilingual dictionary.
2) Dimension Reduction(DR) The method was used to show the effectiveness and informativeness of cross-lingual word embeddings [13, 54]. Suppose we have Uni_Vec as the learned word representation for both source language and target language. Uni_Vec(w) returns the vector for any word w in either language. Instead of using a sparse bag-of-words feature for documents, we represent each document d in both source and target language as

$$\sum_{w \in d} \text{tfidf}(w) \cdot \text{Uni\_Vec}(w)$$

Using the unified representation, a model trained on source data could be directly applied to target data. In our experiment, we implemented this method with the output vector from $RidgeReg$. The classifier was chosen to be an averaged perceptron[2] as used in [13, 54]. We denote this baseline as $DR.RidgeReg$ in the following evaluation.

2.2.3 Empirical Evaluation

Our experiments include the evaluation of the proposed methods in bilingual dictionary extension (as a sub-task), and in cross-lingual text classification as the end-to-end evaluation. We fixed English as the source language. Since it is hard to obtain a large labeled dataset in real low-resource language, we used Spanish, French, German and Chinese available in RCV2[60] to simulate low-resource condition. For completeness, we also include anther smaller internal dataset in Uzbek, a real low-resource language, to prove the effectiveness of our methods.

Evaluation for bilingual dictionary extension

We obtained online bilingual dictionaries from English to Spanish, French and German via *MyMemory* [3] and from English to Chinese via *CC-CEDICT* [4]. The English-Uzbek dictionary was given in the internal dataset. The dictionary sizes are measured using the number of word translation pairs, as summarized in Table 2.1; the branching factor means the average number of translated words in the target language per source word.

Target Language	Size	Branching Factor
Spanish	11518	2.21
French	9901	2.05
German	8856	2.00
Chinese	8185	2.31
Uzbek	9066	2.35

Table 2.1: Statistics of the bilingual dictionaries

[2]We also tried with SVM as a stronger classifier, but it gave similar performance as the averaged perceptron.
[3]http://mymemory.translated.net/
[4]https://www.mdbg.net/chindict/chindict.php?page=cedict

For word embedding in English, we directly used the pre-trained vectors on a Google News dataset[5]. For the other languages, we applied the Continuous Bag-of-Words Model [71, 73] to an unlabeled corpus in each language. Specifically, for Spanish, French, German and Chinese, we used subsets of multilingual Wikipedia pages[6]; for Uzbek, the monolingual text is harvested from the web, which includes news text, blogs, discussion forums, Twitter and reference materials like Wikipedia. Table 2.2 summarizes sizes of these monolingual corpora.

Language	Tokens	Vocabulary Size
English	100B	3M
Spanish	412M	665K
French	488M	754K
German	619M	1505K
Chinese	123M	723K
Uzbek	52M	510k

Table 2.2: The sizes of the monolingual corpora

To simulate the low-resource conditions we sub-sampled 1%, 10%, 25%, 50%, 75% and 100% of the translation pairs in each bilingual dictionary. Specifically, for each fixed percentage we randomly sub-sampled 10 times from the pool, and averaged the performance scores of each method over these ten samples. Each sample was further split into subsets of 50% for training, 25% for validation set (for tuning parameters) and 25% for testing. The hyper-parameters for our method and all baselines are tuned on the validation set.

We evaluated the performance on each test set using a ranking metric. That is, we treated each target-language word in the test set as query, and its true translation in the source language as the relevant items. For each true translation pair, we randomly sampled 100 words in the source language as irrelevant items. The union of all the relevant and irrelevant items of each query form the candidate set for the query. Each candidate was scored by one of the dictionary extension methods (RidgeReg, OrthReg, CCA, TransLP or kNN). By sorting the scores for each query we obtained a ranked list per query. We then evaluated the ranked lists using the mean average precision (MAP), which is conventional in the evaluation of retrieval systems. The MAP scores range from 0 to 1; the higher MAP means the better performance.

Evaluation of the CLTC performance

RCV1/RCV2 Dataset: We used the Reuters RCV1/RCV2 [60] benchmark corpora for this part of the evaluation. RCV1 contains a large number of English news stories and each document belongs to at least one topical category. RCV2 includes news stories in several languages (including the four "low-resource" languages in our study) with topic labels in the same taxonomy. However, the document collections are not parallel, i.e., they are not translations of each other. We used a subset of the English, French and German documents and all the available documents

[5]https://code.google.com/p/word2vec/

[6]https://sites.google.com/site/rmyeid/projects/polyglot#TOC-Download-Wikipedia-Text-Dumps

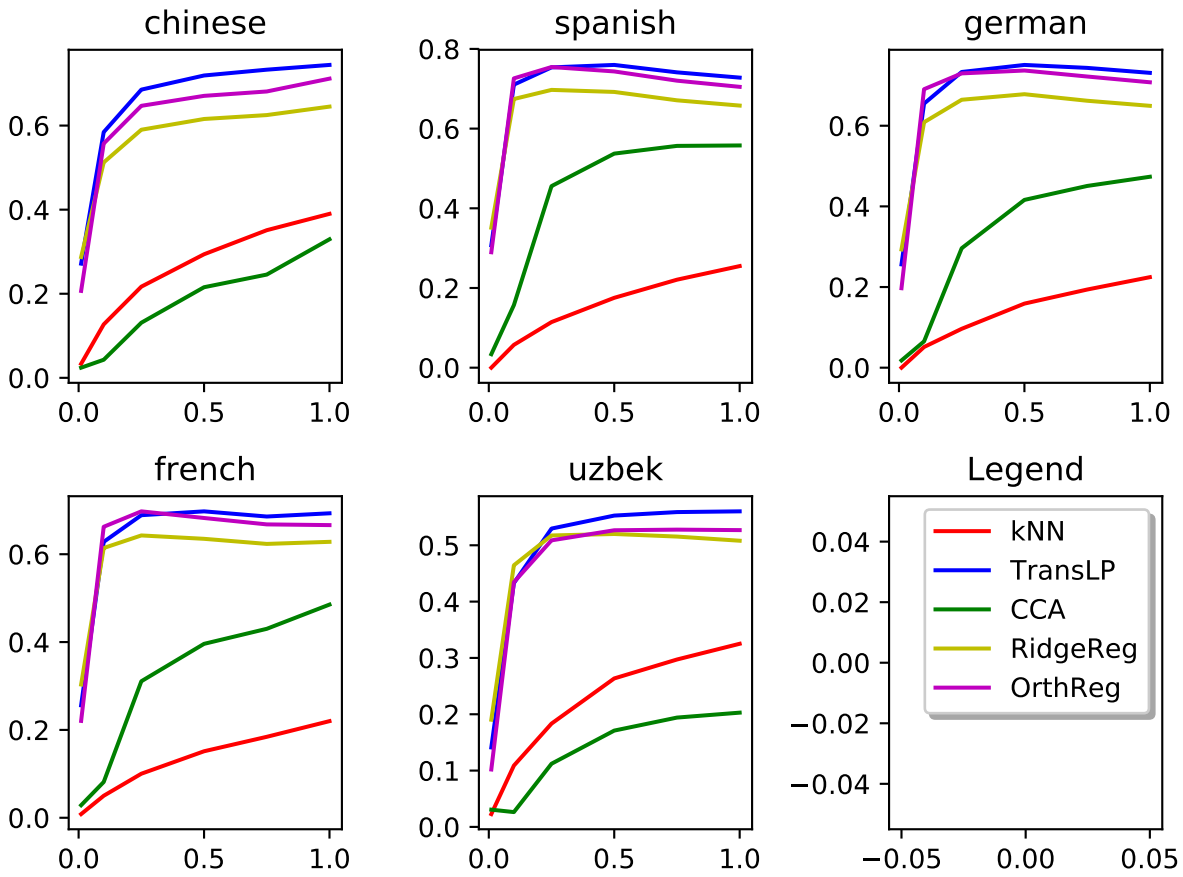


Figure 2.2: Performance (in MAP@5) of our methods in dictionary extension based on multilingual word embeddings

in Spanish and Chinese in our experiments. The statistics of these datasets are shown in table 2.5. All the topic categories in RCV2 are covered by the set of categories contained in RCV1.

The average number of topic labels per document in RCV1 and RCV2 is about 3, thus we have a multi-label classification problem to solve. We trained and translated binary classifier for each individual topic category, and we evaluated the performance in the F_1 measure (micro-averaged and macro-averaged), which has been conventional in text classification [60].

Uzbek Dataset: The internal dataset of Uzbek contains two genres of documents: news articles and discussion forum in both English and Uzbek. Thus we have a binary classification problem. The statistics of each category in English and Uzbek are shown in table 2.6.

Cross Validation

For both RCV1/RCV2 and Uzbek dataset, we used 5-fold cross-validation and split both source and target documents into 5 folds. For each run we trained a classification model with 3 folds of training data in the source language, then we evaluated the cross-lingual classification results

Query	Method	Chinese predictions
kill	TransLP	擇 死 無辜 致死
	OrthReg	殺死 吃掉 死 陷阱
conference	TransLP	會議 談話 輪流 環節
	OrthReg	會 召開 會上 大會
research	TransLP	研究 實驗 科研 評估
	OrthReg	研究 分析 科學 科研
private	TransLP	私人 公有 公用 私有
	OrthReg	私人 公營 私有 公有
style	TransLP	風格 面貌 格局 技法
	OrthReg	風格 樣式 特色 造型

Query	Method	Spanish predictions
kill	TransLP	matar derribar matarlo asesinar
	OrthReg	matar matarlo asesinar destruir
conference	TransLP	conferencia reunión congresos conference
	OrthReg	conferencia reunión conference forum
research	TransLP	investigación investigaciones divulgación científico
	OrthReg	investigación científico investigaciones análisis
private	TransLP	privado privada privados privadas
	OrthReg	privado privada privadas privados
style	TransLP	estilo esquema gusto toque
	OrthReg	estilo gusto estética toque

Table 2.3: Example English words with their closest words in Chinese(ZH), Spanish(ES) using training results from TransLP and OrthReg at 100% size of bilingual dictionary. Blue color in query means there are ground-truth in the bilingual dictionary. Green words in predictions are the correct ones according ground-truth dictionary

Query	Method	German predictions
kill	TransLP	töten ermorden zerstören tötet
	OrthReg	töten tötet ermorden vernichten
conference	TransLP	konferenz tagung konferenzen sitzung
	OrthReg	konferenz tagung gipfeltreffen pressekonferenz
research	TransLP	forschung untersuchungen forschungen untersuchung
	OrthReg	forschung forschungen untersuchungen forschungsergebnisse
private	TransLP	privat private privaten privater
	OrthReg	privat privaten private privater
style	TransLP	stil baustil spielweise musikstil
	OrthReg	stile stil musikstil spielweise

Query	Method	French predictions
kill	TransLP	tuer blesser abattre détruire
	OrthReg	tuer détruire capturer blesser
conference	TransLP	conférence réunion rubrique réunions
	OrthReg	conférence réunion meeting colloque
research	TransLP	recherches recherche bibliographique scientifique
	OrthReg	recherche recherches scientifique chercheur
private	TransLP	privé privées exclusif privés
	OrthReg	privé privées privés privée
style	TransLP	style mode look traditionnel
	OrthReg	style mode look l'esthétique

Table 2.4: Example English words with their closest words in German(DE), French(FR) using training results from TransLP and OrthReg at 100% size of bilingual dictionary. Blue color in query means there are ground-truth in the bilingual dictionary. Green words in predictions are the correct ones according ground-truth dictionary

on one fold of the data (the test set) in the target language. Using the same fold splits, we also trained, validated and tested the classification model directly using the labeled data in the target language, which provides the upper bound performance for model translation. We use MonoTrain to denote this upper bound in section 2.2.3.

Results Analysis

Bilingual Dictionary Extension: Figure 2.2 shows the performance curves of RidgeReg, TransLP, OrthReg, CCA and kNN in dictionary extension for five language pairs, where we used MAP@5 as the metric. For all dictionary sizes and language pairs, TransLP, OrthReg, and RidgeReg substantially outperformed CCA and kNN. For larger dictionary size, TransLP further outperformed RidgeReg and OrthReg. And RidgeReg is usually better for extremely small dictionary size (1% of full size). It is also obvious that the performance of English-Uzbek is worse

	Language	Size	Num. of Categories
RCV1	English	23149	98
RCV2	Spanish	18655	64
	French	20000	70
	German	20000	71
	Chinese	28964	61

Table 2.5: Statistics of RCV1 and RCV2. Size refers to the number of documents. Topic Categories containing less than 5 documents are discarded

Genre	Language	Size
News Article	English	1218
Discussion Forum	English	501
News Article	Uzbek	49893
Discussion Forum	Uzbek	12898

Table 2.6: Statistics of Uzbek dataset. Size refers to the number of documents.

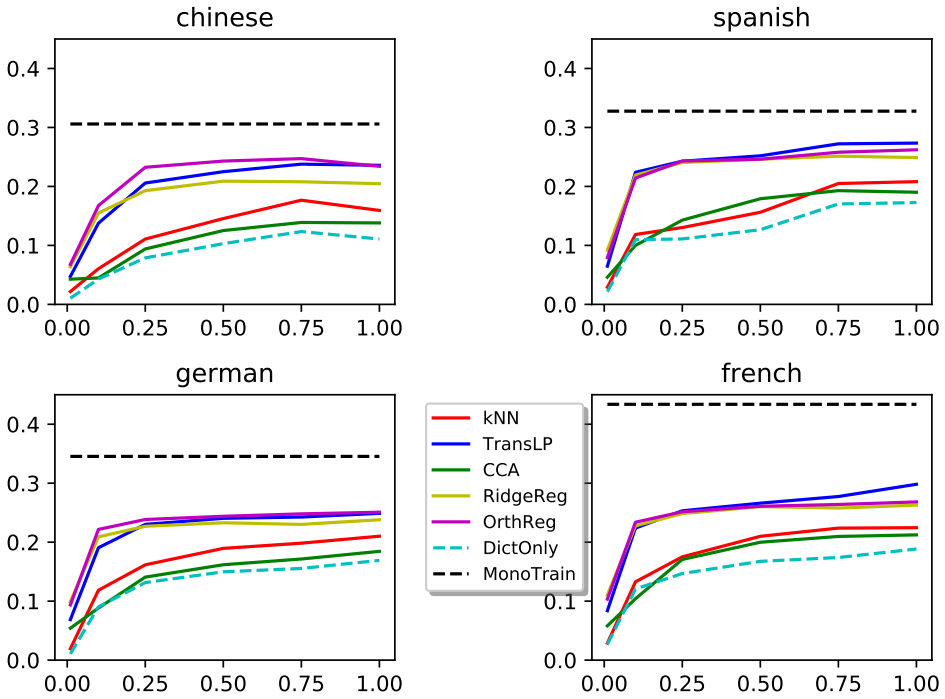


Figure 2.3: Macro-average F1 curves of our CLTC methods for RCV1/RCV2

than four other language pairs. We believe the reason is the relatively smaller size of monolingual corpus (see table 2.2). In table 2.3 and 2.4, we present the system outputs of TransLP

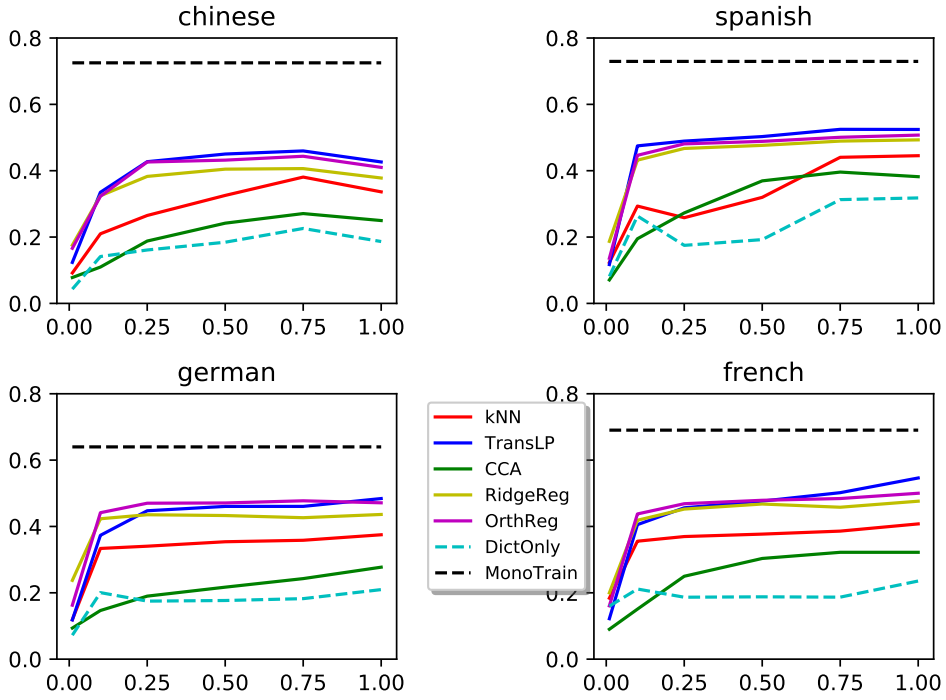


Figure 2.4: Micro-average F1 curves of our CLTC methods for RCV1/RCV2

and OrthReg in 4 target languages. For both methods, the first few extended word pairs are exact translation equivalence in most of the cases. Some predicted translations are very close in meanings, but not identical. For example "style" has been predicted with "baustil" ("architecture" in German). It is also interesting to see sometimes the predicted word pairs captured cross-lingual antonyms, like "公用" ("public" in Chinese) was linked with "private". These results are highly informative for understanding the importance of choosing the right method for bridging language barriers.

CLTC: Figures 2.3 and 2.4 show the end-to-end evaluation results on RCV1/ RCV2 of our proposed methods and baselines in simulated low-resource conditions for the four target languages (Chinese, Spanish, German and French) with shared source language (English). Figure 2.5 shows the result in similar setting for Uzbek dataset. We also include the model translation using non-extended dictionaries, which is named as DictOnly.

Intuitively, the quality of the bilingual dictionary should have a significant impact on CLTC performance. In general, the end-to-end evaluation results are consistent with the ones of dictionary extension. However, there are some inconsistencies in two tasks. For example, for dictionary extension, OrthReg performed better than TransLP at smaller dictionary size but worse at larger dictionary size. However, in German, OrthReg classified documents better than TransLP at all dictionary sizes. A possible reason would be that for dictionary extension, we require extended word pairs to be exact translation equivalence, while in CLTC it is acceptable to have translations with close but not exactly the same meanings.

On RCV1/RCV2, TransLP significantly outperformed RidgeReg, CCA and kNN, showing

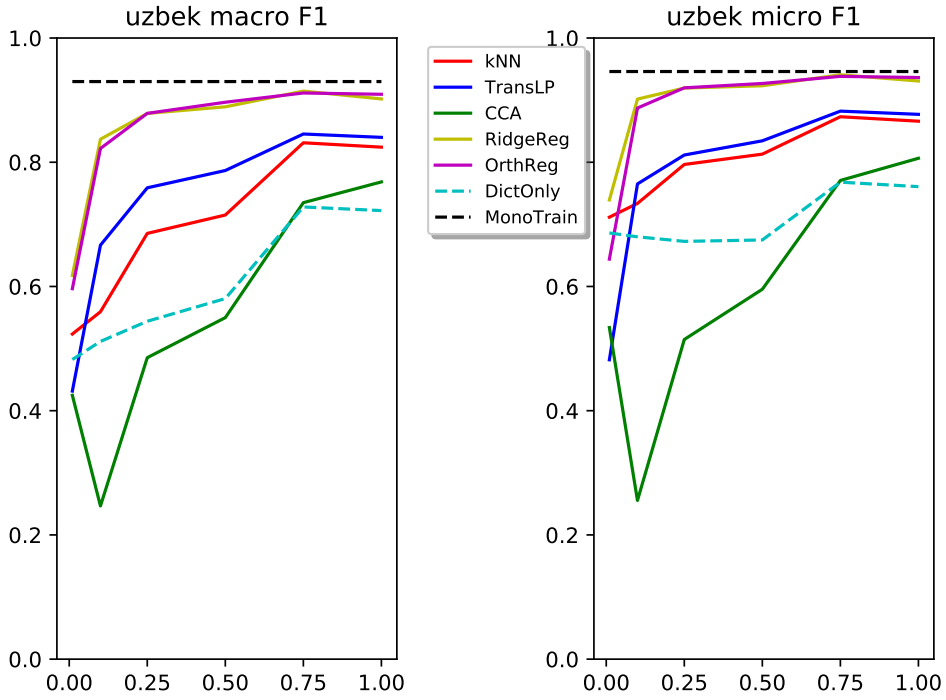


Figure 2.5: Micro-average and Marco-averaged F1 curves of our CLTC methods for Uzbek dataset

our proposed dictionary extension methods are better suitable for CLTC task than the state-of-art (at the time of publication) multilingual word embedding techniques and intuitive heuristic. The performance of TransLP is also comparable with the performance of the newer state-of-art method OrthReg. For the simpler task on Uzbek dataset, RidgeReg and OrthReg outperform all other methods by a large margin. The advantage is more obvious when the dictionary sizes are small.

Another important observation we could draw from both datasets is that CLNB with RidgeReg, TransLP, OrthReg, and kNN all substantially outperformed the results of DictOnly, even when the dictionary is relatively comprehensive. This implies that directly using bilingual dictionaries alone is sub-optimal for CLTC, but using them to establish the mapping among multilingual word embeddings is a winning strategy. For example, with RidgeReg using only 5% to 10% of the dictionaries, we obtained much higher scores in CLTC than that of DictOnly using 100% of the dictionaries in the experiments for all the language pairs.

Table 2.7 compares the performance of our methods (CLNB. RidgeReg, CLNB. TransLP) with that of the method (CLMM) by Shi et al. [98] and the method (DR.RidgeReg) originally proposed by Klementiev et al. and followed by Lauly et al. [13, 54]. The experiments were conducted in two datasets and under the condition that each method used the full-sized bilingual dictionaries (i.e. same cross-lingual knowledge). Recall that CLMM was originally evaluated under the conditions of using full-sized human-defined dictionaries, so we conducted this comparative

Table 2.7: Performance of CLNB, kNN, DictOnly, CLMM, DR using full-sized dictionaries: the results are presented in the format of "Macro-averaged F1/Micro-averaged F1"; bold-face indicates are the best scores for each target language.

Dataset	LANG	CLNB.TransLP	CLNB.RidgeReg	CLMM	DR.RidgeReg	kNN	DictOnly
RCV1/RCV2	Chinese	**0.252**/0.438	0.202/0.399	0.129/0.295	0.059/0.159	0.238/**0.461**	0.098/0.124
	Spanish	0.220/0.506	**0.287**/0.537	0.264/**0.563**	0.099/0.193	0.242/0.475	0.173/0.318
	German	**0.266/0.510**	0.245/0.438	0.220/0.482	0.121/0.295	0.232/0.407	0.169/0.210
	French	**0.267**/0.522	**0.267**/0.487	0.250/**0.553**	0.097/0.286	0.248/0.413	0.188/0.236
Uzbek	Uzbek	0.861/0.886	**0.929/0.950**	0.852/0.877	0.924/0.945	0.885/0.913	0.798/0.803

evaluation under the same condition [7]. We also include the performance of kNN and DictOnly for reference. Both our methods performed comparable or better than CLMM in both datasets and in all language pairs, which shows that although designed for low-resource situations, our proposed methods could reach or beat the performance of state-of-art designed for rich-resource scenarios. The result is not surprising because our methods are capable of utilizing large and general-purpose monolingual corpus in addition to bilingual dictionaries. Another advantage of our methods is efficiency, CLMM is significantly slower than CLNB. TransLP or CLNB. RidgeReg at testing phrase. For CLMM, the prediction time complexity is $\|V\|\times$ (branching factor of dictionary), while the time complexity for CLNB. TransLP and CLNB. RidgeReg is only the averaged number of features (words) for each test data(document). For our setting, CLMM is hundreds of times slower than CLNB, which makes the large-scale evaluation with varying dictionary size intractable for CLMM.

Another interesting comparison is between the different ways of CLTC given multilingual word embeddings. As we could observe from the performance of CLNB.RidgeReg and DR.RidgeReg on RCV1/RCV2, using cross-lingual word similarity to translate the classification model works significantly better than using weighted summation of word vectors as document representation. However, for the simpler task on the Uzbek dataset, the two methods shared similar performance. Therefore, CLNB is always the optimal choice of CLTC, especially for more practical and complicated problems.

2.3 Transfer via parallel corpus (ACL' 17)

Dictionary-based methods including the one we introduced in section 2.2 often ignore the dependency of word meaning and its context, and cannot leverage domain-specific disambiguation when the dictionary on hand is a general-purpose one. Existing parallel-corpus based methods, although more effective in deploying context (when combined with word embedding in particular), often have an issue of domain mismatch or distribution mismatch if the available source-language training data, the parallel corpus (human-aligned or machine-translation induced one) and the target documents of interest are not in the same domain and genre[22]. In this section, we propose a new parallel-corpus based approach, focusing on the reduction

[7]In our experiments CLMM using extended dictionaries had worse results than CLMM when using non-extended dictionaries.

of domain/distribution matches in CLTC. We call this approach Cross-lingual Distillation with Feature Adaptation or CLDFA in short. It is inspired by the recent work in model compression [45] where a large ensemble model is transformed into a compact (small) model. The assumption of knowledge distillation for model compression is that the knowledge learned by the large model can be viewed as a mapping from input space to output (label) space. Then, by training with the soft labels predicted by the large model, the small model can capture most of the knowledge from the large model. Extending this key idea to CLTC, if we see parallel documents as different instantiations of the same semantic concepts in different languages, a target-language classifier should gain the knowledge from a well-trained source classifier by training with the target-language part of the parallel corpus and the soft labels made by the source classifier on the source language side.

More specifically, we propose to distillate knowledge from the source language to the target language in the following 2-step process:

- Firstly, we train a source-language classifier with both labeled training documents and adapt it to the unlabeled documents from the source-language side of the parallel corpus. The adaptation enforces our classifier to extract features that are: 1) discriminative for the classification task and 2) invariant with regard to the distribution shift between training and parallel data.
- Secondly, we use the trained source-language classifier to obtain the *soft* labels for a parallel corpus, and the target-language part of the parallel corpus to train a target classifier, which yields a similar category distribution over target-language documents as that over source-language documents. We also use unlabeled testing documents in the target language to adapt the feature extractor in this training step.

Intuitively, the first step addresses the potential domain/distribution mismatch between the labeled data and the unlabeled data in the source language. The second step addresses the potential mismatch between the target-domain training data (in the parallel corpus) and the test data (not in the parallel corpus). The feature adaptation step makes our framework particularly robust in addressing the distributional difference between in-domain documents and parallel corpus, which is important for the success of CLTC with low-resource languages.

The previous parallel-corpus based CLTC methods [65, 82, 108, 112] learn general cross-lingual features solely from the parallel corpus, which ignores the classification task at hand. In our approach, the target classifier is trained on the soft classification labels from the source classifier. Therefore it can extract task-specific features and produce more accurate predictions.

The main contributions in this section are the following:

- We propose a novel framework (CLDFA) for knowledge distillation in CLTC through a parallel corpus. It can extract task-specific features and closes the domain gap between in-domain documents and parallel corpus.
- CLDFA has the flexibility to be built on a large family of existing monolingual text classification methods. Hence it is very efficient and scalable with the proper choice of plug-in text classifier.
- Our evaluation on benchmark datasets shows that our method had a better or at least comparable performance than that of other state-of-art CLTC methods.

2.3.1 Preliminary

Task and Notation

CLTC aims to use the training data in the source language to build a model applicable in the target language. In our setting, we have labeled data in source language $L_{src} = \{x_i, y_i\}_{i=1}^{L}$, where x_i is the labeled document in source language and y_i is the label vector. We then have our test data in the target language, given by $T_{tgt} = \{x'_i\}_{i=1}^{T}$. Our framework can also use unlabeled documents from both languages in transductive learning settings. We use $U_{src} = \{x_i\}_{i=1}^{M}$ to denote source-language unlabeled documents,$U_{tgt} = \{x'_i\}_{i=1}^{N}$ to denote target-language unlabeled documents, and $U_{parl} = \{(x_i, x'_i)\}_{i=1}^{P}$ to denote a unlabeled bilingual parallel corpus where x_i and x'_i are paired document translations of each other. We assume that the unlabeled parallel corpus does not overlap with the source-language training documents and the target-language test documents.

Convolutional Neural Network (CNN) as a Plug-in Classifier

We use a state-of-the-art CNN-based neural network classifier [52] as the plug-in classifier in our framework. Instead of using a bag-of-words representation for each document, the CNN model concatenates the word embeddings (vertical vectors) of each input document into a $n \times k$ matrix, where n is the length (number of word occurrences) of the document, and k is the dimension of word embedding. Denoting by

$$\mathbf{x_{1:n}} = \mathbf{x_1} \oplus \mathbf{x_2} \oplus ... \oplus \mathbf{x_n}$$

as the resulted matrix, with $\oplus$ the concatenation operator. One-dimensional convolutional filter $\mathbf{w} \in R^{hk}$ with window size h operates on every consecutive h words, with non-linear function f and bias b. For window of size h started at index i, the feature after convolutional filter is given by:

$$c_i = f(\mathbf{w} \cdot \mathbf{x_{i:i+h-1}} + b)$$

A max-over-time pooling [16] is applied on c over all possible positions such that each filter extracts one feature. The model uses multiple filters with different window sizes. The concatenated outputs from filters consist the feature of each document. We can see the convolutional filters and pooling layers as feature extractor $\mathbf{f} = G_f(x, \theta_f)$, where θ_f contains parameters for embedding layer and convolutional layer. Theses features are then passed to a fully connected softmax layer to produce probability distributions over labels. We see the final fully connected softmax layer as a label classifier $G_y(\mathbf{f}, \theta_y)$ that takes the output $\mathbf{f}$ from the feature extractor. The final output of model is given by $G_y(G_f(x, \theta_f), \theta_y)$, which is jointly parameterized by $\{\theta_f, \theta_y\}$

We want to emphasize that our choice of the plug-in classifier here is mainly for its simplicity and scalability to demonstrate our framework. There is a large family of neural classifiers for monolingual text classification that could be used in our framework as well, including other convolutional neural networks by [50], the recurrent neural networks by [18, 51, 56, 102, 128], the attention mechanism by [122], the deep dense network by [48], and more.

2.3.2 Vanilla Distillation

Let us introduce two versions of our model for cross-language knowledge distillation, i.e., the vanilla version and the full version with feature adaptation. Both are supported by the proposed framework. We denote the former by CLD-KCNN and the latter by CLDFA-KCNN.

Without loss of generality, assume we are learning a multi-class classifier for the target language. We have $y \in 1, 2, ..., |v|$ where v is the set of all possible classes. We assume the base classification network produces real number logits q_j for each class. For example, for the case of CNN text classifier, the logits can be produced by a linear transformation which takes features extracted by a max-pooling layer and outputs a vector of size $|v|$. The logits are converted into probabilities of classes through the softmax layer, by normalizing each q_j with all other logits.

$$p_j = \frac{\exp(q_j/T)}{\sum_{k=1}^{|v|} \exp(q_k/T)} \tag{2.25}$$

where T is a temperature and is normally set to 1. Using a higher value of T produces a softer probability distribution over classes.

The first step of our framework is to train the source-language classifier on labeled source documents L_{src}. We use standard temperature $T = 1$ and cross-entropy loss as the objective to minimize. For each example and its label (x_i, y_i) from the source training set, we have:

$$\mathcal{L}(\theta_{src}) = - \sum_{(x_i,y_i)\in L_{src}} \sum_{k=1}^{|v|} \mathbb{1}\{y_i = k\} \log p(y = k|x_i; \theta_{src}) \tag{2.26}$$

where $p(y = k|x; \theta_{src})$ is source model controlled by parameter θ_{src} and $\mathbb{1}\{\cdot\}$ is the indicator function.

In the second step, the knowledge captured in θ_{src} is transferred to the distilled model in the target language by training it on the parallel corpus. The intuition is that paired documents in parallel corpus should have the same distribution of class predicted by the source model and target model. In the simplest version of our framework, for each source-language document in the parallel corpus, we predict a soft class distribution by source model with high temperature. Then we minimize the cross-entropy between soft distribution produced by source model and the soft distribution produced by target model on the paired documents in the target language. More formally, we optimize θ_{tgt} according to the following loss function for each document pair (x_i, x'_i) in parallel corpus.

$$\mathcal{L}(\theta_{tgt}) = - \sum_{(x_i,x'_i)\in U_{parl}} \sum_{k=1}^{|v|} p(y = k|x_i; \theta_{src}) \log p(y = k|x'_i; \theta_{tgt}) \tag{2.27}$$

During distillation, the same high temperature is used for training target model. After it has been trained, we set the temperature to 1 for testing.

We can show that under some assumptions, the two-step cross-lingual distillation is equivalent to distilling a target-language classifier in the target-language input space.

Lemma 1. *Assume the parallel corpus $\{x_i, x_i'\} \in U_{parl}$ is generated by $x_i' \sim p(X'; \eta)$ and $x_i = t(x_i')$, where η controls the marginal distribution of x_i and t is a differentiable translation function with integrable derivative. Let $f_{\theta_{src}}(t(x'))$ be the function that outputs soft labels of $p(y = k|t(x'); \theta_{src})$. The distillation given by equation 2.27 can be interpreted as distillation of a target language classifier $f_{\theta_{src}}(t(x'))$ on target language documents sampled from $p(X'; \eta)$.*

$f_{\theta_{src}}(t(x'))$ is the classifier that takes input of target documents, translates them into source documents through t and makes prediction using the source classifier. If we further assume the testing documents have the same marginal distribution $P(X'; \eta)$, then the distilled classifier should have similar generalization power as $f_{\theta_{src}}(t(x'))$.

Theorem 2. *Let source training data $x_i \in L_{src}$ has marginal distribution $p(X; \lambda)$. Under the assumptions of lemma 1, further assume $p(t(x'); \lambda) = p(x'; \eta)$, $p(y|t(x')) = p(y|x')$ and $t'(x') \approx C$, where C is a constant. Then $f_{\theta_{src}}(t(x'))$ actually minimizes the expected loss in target language data $E_{x' \sim p(X;\eta), y \sim p(Y|x')}[L\big(y, f(t(x'))\big)]$.*

Proof. By definition of equation 2.26, $f_{\theta_{src}}(x)$ minimizes the expected loss $E_{x \sim p(X;\lambda), y \sim p(Y|x)}[L\big(y, f(x)\big)]$, where L is cross-entropy loss in our case. Then we can write

$$
\begin{aligned}
& E_{x \sim p(X;\lambda), y \sim p(Y|x)}[L\big(y, f(x)\big)] \\
=& \int p(x; \lambda) \sum_y p(y|x) L\big(y, f(x)\big) dx \\
=& \int p(t(x'); \lambda) \sum_y p(y|t(x')) L\big(y, f(t(x'))\big) t'(x') dx' \\
\approx& C \int p(x'; \eta) \sum_y p(y|x') L\big(y, f(t(x'))\big) dx' \\
=& C E_{x' \sim p(X;\eta), y \sim p(Y|x')}[L\big(y, f(t(x'))\big)]
\end{aligned}
$$

□

2.3.3 Distillation with Adversarial Feature Adaptation

Although vanilla distillation is intuitive and simple, it cannot handle distribution mismatch issues. For example, the marginal feature distributions of source-language documents in L_{src} and U_{parl} could be different, so are the distributions of target-language documents in U_{parl} and T_{tgt}. According to theorem 2, the vanilla distillation works for the best performance under unrealistic assumption: $p(t(x')|\lambda) = p(x'|\eta)$. To further illustrate our point, we trained a CNN classifier according to equation 2.26 and used the features extracted by G_f to present the source-language documents in both L_{src} and U_{parl}. Then we projected the high-dimensional features onto a 2-dimensional space via t-Distributed Stochastic Neighbor Embedding (t-SNE)[90]. This resulted the visualization of the project data in Figures 2.6 and 2.7.

It is quite obvious in Figure 2.6 that the general-purpose parallel corpus has a very different feature distribution from that of the labeled source training set. Even for machine-translated parallel data from the same domain, as shown in figure 2.7, there is still a non-negligible distribution shift from the source language to the target language for the extracted features. Our

interpretation of this observation is that when the MT system (e.g. Google Translate) is a general-purpose one, it non-avoidably add translation ambiguities which would lead the distribution shift from the original domain. To address the distribution divergence brought by either a general-purpose parallel corpus or an imperfect MT system, we seek to adapt the features extraction part of our neural classifier such that the feature distributions on both sides should be close as possible in the newly induced feature space. We adapt the adversarial training method by [27] to the cross-lingual settings in our problems.

Given a set of training set of $L = \{x_i, y_i\}_{i=1,...,N}$ and an unlabeled set $U = \{x'_i\}_{i=1,...,M}$, our goal is to find a neural classifier $G_y(G_f(x, \theta_f), \theta_y)$, which has good discriminative performance on L and also extracts features which have similar distributions on L and U. One way to maximize the similarity of two distributions is to maximize the loss of a discriminative classifier whose job is to discriminate the two feature distributions. We denote this classifier by $G_d(\cdot, \theta_d)$, which is parameterized by θ_d.

At training time, we seek θ_f to minimize the loss of G_y and maximize the loss of G_d. Meanwhile, θ_y and θ_d are also optimized to minimize their corresponding loss. The overall optimization could be summarized as follows:

$$
\begin{aligned}
E(\theta_f, \theta_y, \theta_d) = & \sum_{x_i, y_i \in L} L_y(y_i, G_y(G_f(x_i, \theta_f), \theta_y)) \\
& - \alpha \sum_{x_i \in L} L_d(0, G_d(G_f(x_i, \theta_f), \theta_d)) \\
& - \alpha \sum_{x_j \in U} L_d(1, G_d(G_f(x_j, \theta_f), \theta_d))
\end{aligned}
$$

where L_y is the loss function for true labels y, L_d is loss function for binary labels indicating the source of data and α is the hyperparameter that controls the relative importance of two losses. We optimize θ_f, θ_y for minimizing E and optimize θ_d for maximizing E. We jointly optimize $\theta_f, \theta_y, \theta_d$ through the gradient reversal layer[27].

We use this feature adaptation technique to firstly adapt the source-language classifier to the source-language documents of the parallel corpus. When training the target-language classifier by matching soft labels on the parallel corpus, we also adapt the classifier to the target testing documents. We use cross-entropy loss functions as L_y and L_d for both feature adaptation.

2.3.4 Empirical Evaluation

Dataset

Our experiments used two benchmark datasets, as described below.

(1) Amazon Reviews

We used the multilingual multi-domain Amazon review dataset created by Prettenhofer and Stein [84]. The dataset contains Amazon reviews in three domains: book, DVD and music. Each domain has reviews in four different languages: English, German, French, and Japanese. We treated English as the source language and the rest three as the target languages, respectively.

Language	Domain	# of Documents
English	book	50000
	DVD	30000
	music	25220
German	book	165470
	DVD	91516
	music	60392
French	book	32870
	DVD	9358
	music	15940
Japanese	book	169780
	DVD	68326
	music	55892

Table 2.8: Dataset Statistics for the Amazon reviews dataset

This gives us 9 tasks (the product of the 3 domains and the 3 target languages) in total. For each task, there are 1000 positive and 1000 negative reviews in English and the target language, respectively. [84] also provides 2000 parallel reviews per task, that were generated using Google Translate [8], and used by us for cross-language distillation. There are also several thousands of unlabeled reviews in each language. The statistics of unlabeled data is summarized in Table 2.8. All the reviews are tokenized using standard regular expressions except for Japanese, for which we used a publicly available segmenter [9].

(2) English-Chinese Yelp Hotel Reviews This dataset was firstly used for CLTC by [15]. The task is to make sentence-level sentiment classification with 5 labels(rating scale from 1 to 5), using English as the source language and Chinese as the target language. The labeled English data consists of balanced labels of 650k Yelp reviews from Zhang et al. [129]. The Chinese data includes 20k labeled Chinese hotel reviews and 1037k unlabeled ones from [62]. Following the approach by [15], we use 10k of labeled Chinese data as the validation set and another 10k hotel reviews as held-out test data. To avoid the performance gap caused by different word embedding initialization, we used the same bilingual word embedding for initialization as in [15] and a random sample of 500k parallel sentences from UM-corpus[103], which is a general-purpose corpus designed for machine translation.

Baselines

We compare the proposed method with other state-of-the-art methods as outlined below.

(1) Parallel-Corpus based CLTC Methods Methods in this category all use an unlabeled

[8] translate.google.com

[9] https://pypi.python.org/pypi/tinysegmenter

Target Language	Domain	PL-LSI	PL-KCCA	PL-OPCA	PL-MC	CLD-KCNN	CLDFA-KCNN
German	book	77.59	79.14	74.72	79.22	82.54	**83.95***
	DVD	79.22	76.73	74.59	81.34	82.24	**83.14***
	music	73.81	79.18	74.45	**79.39**	74.65	79.02
French	book	79.56	77.56	76.55	81.92	81.60	**83.37**
	DVD	77.82	78.19	70.54	81.97	82.41	**82.56**
	music	75.39	78.24	73.69	79.30	83.01	**83.31***
Janpanese	book	72.68	69.46	71.41	72.57	74.12	**77.36***
	DVD	72.55	74.79	71.84	76.60	79.67	**80.52***
	music	73.44	73.54	74.96	76.21	73.69	**76.46**
Averaged Accuracy		75.78	76.31	73.64	78.72	79.33	**81.08***

Table 2.9: Accuracy scores of methods on the Amazon Reviews dataset: the best score in each row (a task) is highlighted in bold face. If the score of CLDFA-KCNN is statistically significantly better (in one-sample proportion tests) than the best among the baseline methods, it is marked using a star.

Model	Accuracy
mSDA	31.44%
MT-LR	34.01%
MT-DAN	39.66%
ADAN	41.04%
CLD-KCNN	40.96%
CLDFA-KCNN	**41.82%**

Table 2.10: Accuracy scores of methods on the English-Chinese Yelp Hotel Reviews dataset

parallel corpus. Methods named **PL-LSI** [65], **PL-OPCA** [82] and **PL-KCAA** [108] learn latent document representations in a shared low-dimensional space by performing the Latent Semantic Indexing (LSI), the Oriented Principal Component Analysis (OPCA) and a kernel (namely KCAA) for the parallel text. **PL-MC** [112] recovers missing features via Matrix Completion, and also uses LSI to induce a latent space for parallel text. All these methods train a classifier in the shared feature space with labeled training data from both the source and target languages.

(2) MT-based CLTC Methods The methods in this category all use an MT system to translate each test document in the target language to the source language in the testing phase. The prediction on each translated document is made by a source-language classifier, which can be a Logistic Regression model (**MT+LR**) [15] or a deep averaging network (**MT+DAN**) [15].

(3) Adversarial Deep Averaging Network Similar to our approach, the adversarial Deep Averaging Network (**ADAN**) also exploits adversarial training for CLTC [15]. However, it does not have the parallel-corpus based knowledge distillation part (which we do). Instead, it uses averaged bilingual embeddings of words as its input and adapts the feature extractor to produce similar features in both languages.

We also include the results of **mSDA** for the Yelp Hotel Reviews dataset. **mSDA** [14] is a domain adaptation method based on stacked denoising autoencoders, which has been proved to be effective in cross-domain sentiment classification evaluations. We show the results reported

by [14], where they used bilingual word embedding as input for mSDA.

Implementation Detail

We pre-trained both the source and target classifier with unlabeled data in each language. We ran word2vec[71] [10] on the tokenized unlabeled corpus. The learned word embeddings are used to initialize the word embedding lookup matrix, which maps input words to word embeddings and concatenates them into an input matrix.

We fine-tuned the source-language classifier on the English training data with 5-fold cross-validation. For English-Chinese Yelp-hotel review dataset, the temperature T(Section 2.3.2) in distillation is tuned on validation set in the target language. For Amazon review dataset, since there is no default validation set, we set temperature from low to high in $\{1, 3, 5, 10\}$ and take the average among all predictions.

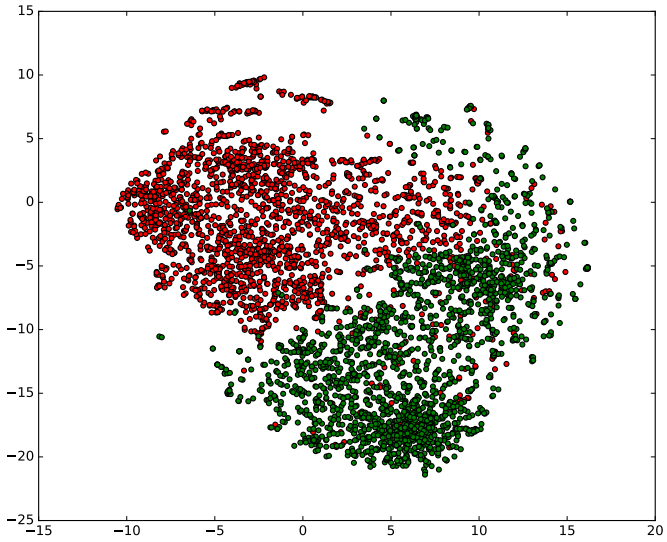
Main Results

In tables 2.9 and 2.10 we compare the results of our methods (the vanilla version CLD-KCNN and the full version CLDFA-KCNN) with those of other methods based on the published results in the literature. The baseline methods are different in these two tables as they were previously evaluated (by their authors) on different benchmark datasets. Clearly, CLDFA-KCNN outperformed the other methods on all except one task in these two datasets, showing that knowledge distillation is successfully carried out in our approach.

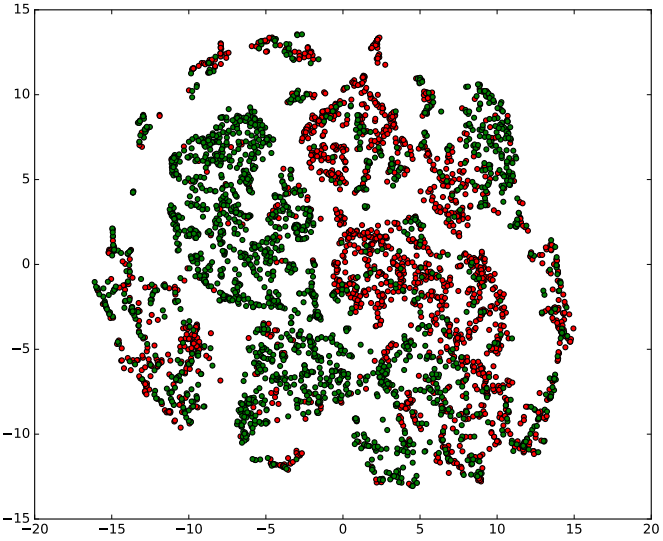
Noticing that CLDFA-KCNN outperformed CLD-KCNN, showing the effectiveness of adversarial feature extraction in reducing the distribution mismatch between the parallel corpus and the train/test data in the target domain. The visualization of extracted features in figure 2.7 and figure 2.6 also shows that CLDFA-KCNN is able to extract more domain-invariant features comparing with CLD-KCNN. We should also point out that in Table 2.9, the four baseline methods (PL-LSI, PL-KCCA, PL-OPCA and PL-MC) were evaluated under the condition of using additional 100 labeled target documents for training, according to the author's report [112]. On the other hand, our methods (CLD-KCNN and CLDFA-KCNN) were evaluated under a tougher condition, i.e., not using any labeled data in the target domains.

We also test our framework when a few training documents in the target language are available. A simple way to utilize the target-language supervision is to fit the target-language model with labeled target data after optimizing with our cross-lingual distillation framework. The performance of CLD-KCNN and CLDFA-KCNN trained with different sizes of labeled target-language data is shown in figure 2.8. We also compare the performance of training the same classifier using only the target-language labels(**Target Only** in figure 2.8). As we can see, our framework can efficiently utilize the extra supervision and improve the performance over the training using only the target-language labels. The margin is most significant when the size of the target-language label is relatively small.

[10] https://code.google.com/archive/p/word2vec/

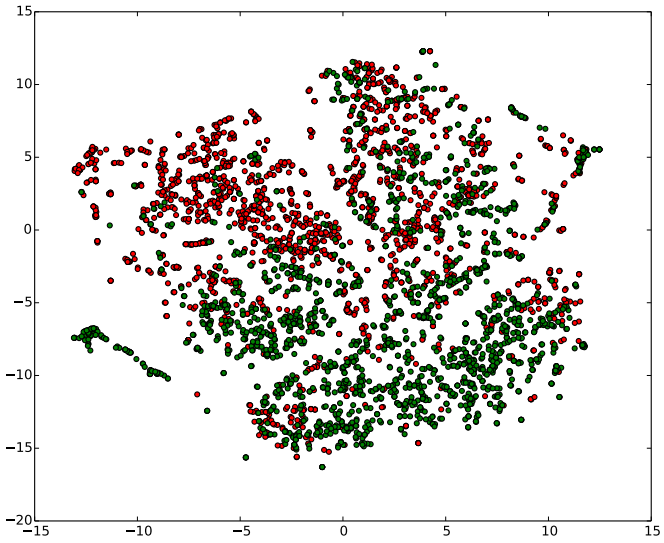


(a) Yelp: w/o. feature adaptation

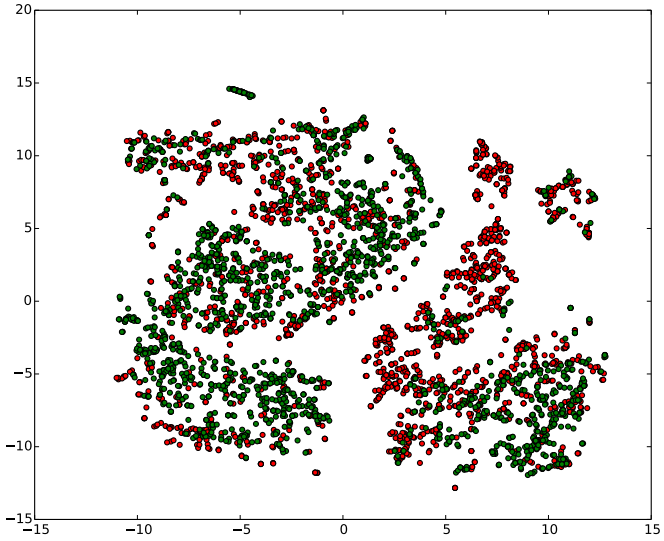


(b) Yelp: w/ feature adaptation

Figure 2.6: Extracted features for source-language documents in the English-Chinese Yelp Hotel Review dataset. Red dots represent features of the documents in L_{src} and green dots represent the features of documents in U_{parl}, which is a general-purpose parallel corpus. The top one is the feature from CLD-KCNN and the bottom one is from CLDFA-KCNN.



(a) Japanese-Music: w/o feature adaptation



(b) Japanese-Music: w/ feature adaptation

Figure 2.7: Extracted features for source-language documents in Japanese split of the Amazon Reviews dataset. Red dots represent features of the documents in L_{src} and green dots represent the features of documents in U_{parl}, which are the machine-translated documents from a target language. The top one is features from CLD-KCNN and the bottom one is from CLDFA-KCNN.

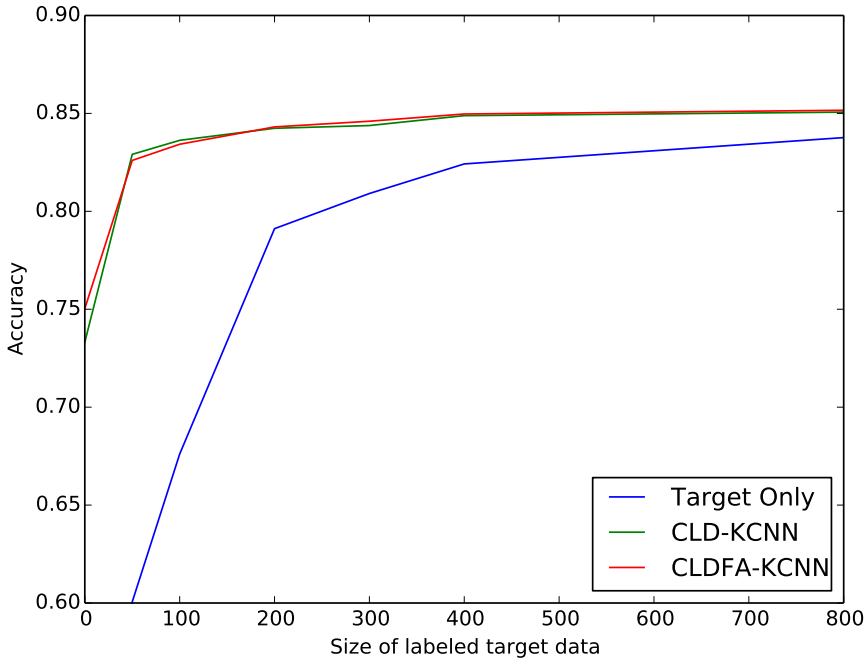


Figure 2.8: Accuracy scores of methods using varying sizes of target-language labeled data on the Amazon review dataset. The target language is German and the domain is music. The parallel corpus has a fixed size of 1000 and the size of the labeled target-language documents is shown on the x-axis

Chapter 3

Cross-lingual Transfer of Word Embeddings

3.1 Motivation

Although the text classification model could be efficiently and effectively transferred across different languages using the techniques we introduced in chapter 2. The technique may not directly apply to other natural language tasks. For instance, it is not straightforward for RidgeReg and TransLP to work on phrase-level and sentence-level features. It is also difficult to extend CLD and CLDFA for word-level tasks(e.g. POS-tagging, Named Entity Recognition). To make the domain transfer more broadly applicable for different tasks, we investigate the cross-lingual transfer of word embeddings, which are common input features for a variety of neural NLP models.

Word embeddings are well known to capture meaningful representations of words based on large text corpora [71, 80]. Training word vectors using monolingual corpora is a common practice in various NLP tasks. However, how to establish cross-lingual semantic mapping among monolingual embeddings remain an open challenge as the availability of resources and benchmarks are highly imbalanced across languages.

Recently, an increasing effort of research has been motivated to address this challenge. Successful cross-lingual word mapping will benefit many cross-lingual learning tasks, such as to transform text classification models trained in resource-rich languages to low-resource languages. Downstream applications include word alignment, text classification, named entity recognition, dependency parsing, POS-tagging, and more [101, 113]. Early works of cross-lingual transfer of word embeddings are based on supervised or semi-supervised learning, i.e., they require cross-lingual supervision such as human-annotated bilingual lexicons and parallel corpora [5, 67, 100].

However, such a requirement may not be met for many language pairs in the real world. Therefore, more recent methods [7, 58, 126] focus on the unsupervised setting, which requires zero cross-lingual supervision. In spite of their promising performance, the previous unsupervised models are usually evaluated in a favorable setting and are not robust for tasks between challenging language pairs. To resolve this issue, we propose a more robust unsupervised

method and extensively evaluated it on various language pairs. In the remaining of the chapter, we will first review supervised methods in section 3.2 and introduce our novel unsupervised model in section 3.3.

3.2 Supervised Cross-lingual Word Embedding

There is a rich body of supervised methods for learning cross-lingual transfer of word embeddings based on bilingual dictionaries [5, 23, 24, 31, 71, 114], sentence-aligned corpora[32, 43, 55] and document-aligned corpora[101, 110]. One of the very first works is that by Mikolov et al. [71] where they showed monolingual word embeddings are likely to share similar geometric properties across languages although they are trained separately and hence cross-lingual mapping can be captured by a linear transformation across embedding spaces. Several follow-up studies tried to improve the cross-lingual transformation in various ways [3, 5, 5, 23, 24, 99, 114, 131]. Nevertheless, all these methods require bilingual lexicons for supervised learning. Vulić and Korhonen [109] showed that 5000 high-quality bilingual lexicons are sufficient for learning a reasonable cross-lingual mapping.

3.3 Unsupervised Transfer(EMNLP'18)

In this section, we propose an unsupervised approach to the cross-lingual transfer of monolingual word embeddings, which requires zero cross-lingual supervision.

Earlier work simply relied on word occurrence information only [26, 89] while later efforts have considered more sophisticated statistics in addition [40]. The main difficulty in unsupervised learning of cross-lingual mapping is the formulation of the objective function, i.e., how to measure the goodness of an induced mapping without any supervision is a non-trivial question. Cao et al. [12] tried to match the mean and standard deviation of the embedded word vectors in two different languages after mapping the words in the source language to the target language. However, such an approach has shown to be sub-optimal because the objective function only carries the first and second order statistics of the mapping. Artetxe et al. [6] tried to impose an orthogonal constraint to their linear transformation model and minimize the distance between the transferred source-word embedding and its nearest neighbor in the target embedding space. Their method, however, requires a seed bilingual dictionary as the labeled training data and hence is not fully unsupervised. [70, 126] adapted a generative adversarial network (GAN) to make the transferred embedding of each source-language word indistinguishable from its true translation in the target embedding space [30]. The adversarial model could be optimized in a purely unsupervised manner but is often suffered from unstable training, i.e. the adversarial learning does not always improve the performance over simpler baselines. Zhang et al. [127], Lample et al. [58] and Artetxe et al. [6] also tried adversarial approaches for the induction of seed bilingual dictionaries, as a sub-problem in the cross-lingual transfer of word embedding.

More recent work [1] improves the performance and robustness by stochastic self-learning [8], maximizing mean discrepancy [120], enforcing cycle consistency [75] and normalizing

[1] after the submission of our work [118]

flows[133].

The key idea of our proposed method is to optimize the mapping in both directions for each language pair (say A and B), in the way that the word embedding translated from language A to language B will match the distribution of word embedding in language B. And when translated back from B to A, the word embedding after two steps of transfer will be maximally close to the original word embedding. A similar property holds for the other direction of the loop (from B to A and then from A back to B).

Specifically, we use the Sinkhorn distance [17] to capture the distributional similarity between two sets of embeddings after transformation, which we found empirically superior to the KL-divergence [126] and distance to nearest neighbor [6, 58] with regards to the quality of learned transformation as well as the robustness under different training conditions.

Among all related works, the closest one to our method is the model proposed in [127], which also uses the Sinkhorn solver inside their self-training process. For each iteration of self-training, the current cross-lingual mapping is used to infer the cross-lingual dictionary, which is then used to train cross-lingual mapping of the next-step. The iteration stops until some convergence criteria are met. In our method, we use the Sinkhorn distance as part of our training objective and optimize it using standard mini-batch stochastic gradient descent with back-propagation. Another difference is that our method optimizes the transformations of both directions between two languages, whereas [127] only optimizes for one direction.

Our novel contributions in the proposed work include:

- We propose an unsupervised learning framework which incorporates the Sinkhorn distance as a distributional similarity and optimizes it in an end-to-end manner.
- Unlike previous models which only consider cross-lingual transformation in a single direction, our model jointly learns the word embedding transfer in both directions for each language pair.
- We present an intensive comparative evaluation where our model achieved state-of-the-art performance for many language pairs in cross-lingual tasks.

3.3.1 Proposed Method

Our system takes two sets of monolingual word embeddings of dimension d as input, which are trained separately on two languages. We denote them as $X = \{x_i\}_{i=1}^{n}$, $Y = \{y_j\}_{j=1}^{m}$, $x_i, y_j \in \mathbb{R}^d$. During the training of monolingual word embedding for X and Y, we also have the access to the word frequencies, represented by vectors $r \in \mathbb{N}^n$ and $c \in \mathbb{N}^m$ for X and Y, respectively. Specifically, r_i is the frequency for word (embedding) x_i and similarly for c_j of y_j. As illustrated in Figure 3.3.1, our model has two mappings: $G : X \to Y$ and $F : Y \to X$. We further denote transferred embedding from X as $G(X) := \{G(x_i)\}_{i=1}^{n}$ and correspondingly for $F(Y)$.

In the unsupervised setting, the goal is to learn the mapping G and F without any paired word translation. To achieve this, our loss function consists of two parts: Sinkhorn distance[17] for matching the distribution of transferred embedding to its target embedding distribution; and a back-translation loss for preventing degenerated transformation.

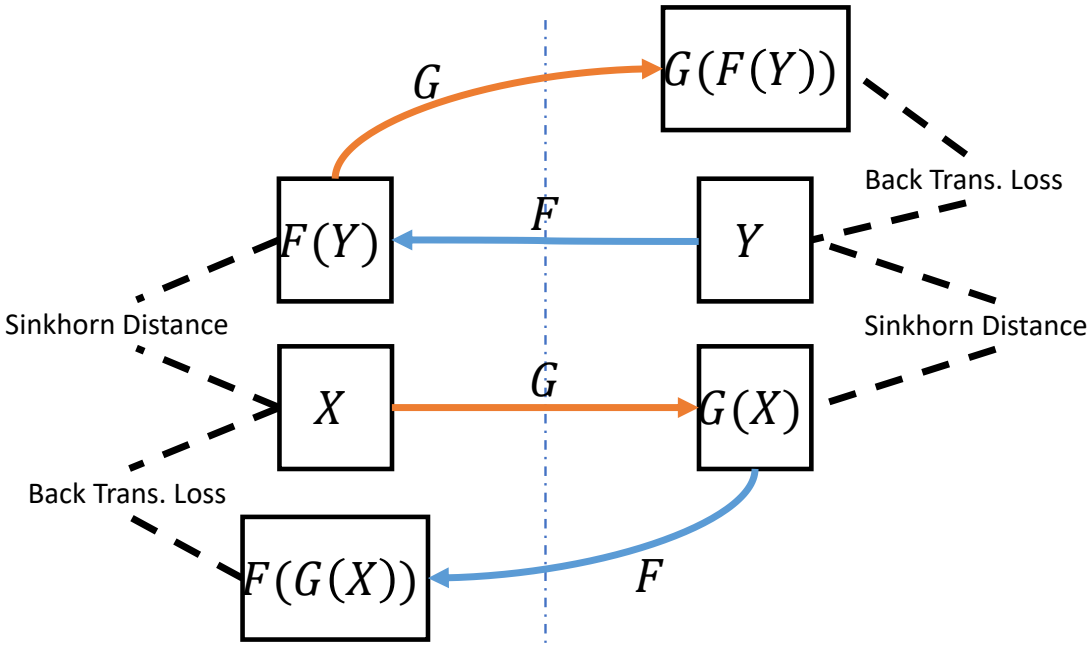


Figure 3.1: The model takes monolingual word embedding X and Y as input. G and F are embedding transfer functions parameterized by a neural network, which are represented by solid arrows. The dashed lines indicate the input for our objective losses, namely the Sinkhorn distance and back-translation loss

.

Sinkhorn Distance

Definition Sinkhorn distance is a recently proposed distance between probability distributions. We use the Sinkhorn distance to measure the closeness between $G(X)$ and Y, and also between $F(Y)$ and X. During the training, our model optimizes G and F for lower Sinkhorn distance to make the transferred embeddings match the distribution of the target embeddings. Here we only illustrate the Sinkhorn distance between $G(X)$ and Y, the derivation for $F(Y)$ and X is very similar. Although the vocabulary sizes of two languages could be different, we can sample mini-batches of equal size from $G(X)$ and Y. therefore we assume $n = m$ in the following derivation.

To compute Sinkhorn distance, we firstly compute a distance matrix $M^{(G)} \in \mathbb{R}^{n \times m}$ between $G(X)$ and Y where $M_{ij}^{(G)}$ is the distance measure between $G(x_i)$ and y_j. The superscript on $M^{(G)}$ indicates the distance that depends on a parameterized transformation G. For instance, if we choose Euclidean distance as a measure (see subsection 3.3.1 for more discussions), we will

Algorithm 1 Computation of Sinkhorn Distance $d_{sh}(G)$

procedure SINKHORN($M^{(G)}, r, c, \lambda, I$)
  $K^{(G)} := e^{-\lambda M^{(G)}}$
  $v = \mathbb{1}_m/m$ ▷ normalized one vector
  $i = 0$
  while $i < I$ **do** ▷ iterate for I times
    $u = r./K^{(G)}v$
    $v = c./{K^{(G)}}^T u$
    $i = i + 1$
  $d_{sh}(G) = u^T((K^{(G)} \otimes M^{(G)})v)$
  return $d_{sh}(G)$ ▷ The Sinkhorn distance

have

$$M_{ij}^{(G)} = \|G(x_i) - y_j\|_2.$$

Given the distance matrix, the Sinkhorn distance between $P_{G(X)}$ and P_Y is defined as:

$$d_{sh}(G) := \min_{P \in U_\alpha(r,c)} \langle P, M^{(G)} \rangle \tag{3.1}$$

where $\langle \cdot, \cdot \rangle$ is the Forbenius dot-product and $U_\alpha(r, c)$ is an entropy constrained transport polytope, defined as

$$\begin{aligned} U_\alpha(r, c) = \{P \in {\mathbb{R}_+}^{n \times m} | P\mathbb{1}_m = r, P^T\mathbb{1}_n = c, \\ h(P) \leq h(r) + h(c) - \alpha\} \end{aligned} \tag{3.2}$$

Note that P is non-negative and the first two constraints make its element-wise sum be 1. Therefore, P can be seen as a set of probability distributions. The same applies for r and c since they are frequencies. h is the entropy function defined on any probability distributions and α is a hyperparameter to choose. For any probabilistic matrix $P \in U_\alpha(r, c)$, it can be viewed as the joint probability of $(G(X), Y)$. The first two constraints ensure that P has marginal distribution on $G(X)$ as $P_{G(X)}$ and on Y as P_Y. We can also view P_{ij} as the evidence for establishing a translation between word vector x_i and word vector y_j.

An intuitive interpretation of equation (3.1) is that we are trying to find the optimal transport probability P under the entropy constraint such that the total distance to transport from $G(X)$ to Y is minimized.

Computing Sinkhorn Distance $d_{sh}(G)$ Cuturi [17] showed that the optimal solution of formula (3.1) has the form $P^* = \mathbf{diag}(u)K\mathbf{diag}(v)$, where u and v are some non-negative vectors and $K^{(G)} := e^{-\lambda M^{(G)}}$; λ is the Lagrange multiplier for the entropic constraint in 3.2 and each α in Equation (3.1) has one corresponding λ. The Sinkhorn distance can be efficiently computed by a matrix scaling algorithm. We present the pseudo code in Algorithm 1. Note that the computation of $d_{sh}(G)$ only requires matrix-vector multiplication. Therefore, we can compute and backpropagate the gradient of $d_{sh}(G)$ with regards to the parameters in G using standard deep learning libraries. We show our implementation details in subsection 3.3.1 and supplementary material.

Choice of the Distance Metric In subsection 3.3.1, we used the Euclidean distance of vector pairs to define $M^{(G)}$ and Sinkhorn distance $d_{sh}(G)$. However, in our preliminary experiment, we found that the Euclidean distance of unnormalized vectors gave poor performance. Therefore, following the common practice, we normalize all word embedding vectors to have a unit L2 norm in the construction of $M^{(G)}$.

As pointed out in Theorem 1 of Cuturi [17], $M^{(G)}$ must be a valid metric in order to make $d_{sh}(G)$ a valid metric. For example, the commonly used cosine distance, which is defined as $\text{CosDist}(a, b) = 1 - \cos(a, b)$, is not a valid metric because it does not satisfy triangle inequality [2]. Thus, for constructing $M^{(G)}$, we propose the square root cosine distance (SqrtCosDist) below:

$$\text{SqrtCosDist}(a, b) := \sqrt{2 - 2\cos(a, b)} \tag{3.3}$$

$$M_{ij}^{(G)} = \text{SqrtCosDist}(G(x_i), y_j) \tag{3.4}$$

Theorem 3. *SqrtCosDist is a valid metric.*

Proof. $\forall a, b \in \mathbb{R}^d$, let $\hat{a} = \frac{a}{\|a\|}$, $\hat{b} = \frac{b}{\|b\|}$. We have $\cos(a, b) = \langle \hat{a}, \hat{b} \rangle$ and $\langle \hat{a}, \hat{a} \rangle = \langle \hat{b}, \hat{b} \rangle = 1$. Then

$$\begin{aligned}
&\text{SqrtCosDist}(a, b) = \sqrt{2 - 2\cos(a, b)} \\
&= \sqrt{\langle \hat{a}, \hat{a} \rangle + \langle \hat{b}, \hat{b} \rangle - 2\langle \hat{a}, \hat{b} \rangle} \\
&= \sqrt{\langle \hat{a} - \hat{b}, \hat{a} - \hat{b} \rangle} \\
&= \|\hat{a} - \hat{b}\|
\end{aligned}$$

Obviously, the last term is the Euclidean distance between normalized input vectors $\hat{a}$ and $\hat{b}$. Since Euclidean distance is a valid metric, it follows that SqrtCosDist satisfies all the axioms for a valid metric. □

Objective Function

Given enough capacity, G is capable to transfer X to Y for arbitrary word-to-word mappings. To ensure that, we learn a meaningful translation and also to regularize the search space of possible transformations, we enforce the word embedding after the forward and the backward transformation should not diverge much from its original direction. We simply choose the back-translation loss based on the cosine similarity:

$$\begin{aligned}
d_{bt}(G, F) = &\sum_i 1 - \cos(x_i, F(G(x_i))) + \\
&\sum_j 1 - \cos(y_i, G(F(y_i)))
\end{aligned} \tag{3.5}$$

where cos is the cosine similarity.

[2] If we select $a = [1, 0], b = [\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}], c = [0, 1]$ We have $\text{CosDist}(a, c) \geq \text{CosDist}(a, b) + \text{CosDist}(b, c)$, which violates the triangle inequality.

Figure 3.2: Different training objectives(y-axes) w.r.t training steps(x-axis).

Putting everything together, we minimize the following objective function.

$$L_{X,Y,r,c}(G, F) = d_{sh}(G) + d_{sh}(F) + \beta d_{bt}(G, F) \tag{3.6}$$

where hyper-parameter β controls the relative weight of the last term against the first two terms in the objective function. By definition, computation of $d_{sh}(G)$ or $d_{sh}(F)$ involves another minimization problem as shown in Equation (3.1). We solve it using the matrix scaling algorithm in subsection 3.3.1, and treat $d_{sh}(G)$ as a deterministic and differentiable function of parameters in G. The same holds for $d_{sh}(F)$ and F.

Wasserstein GAN Training for Good Initial Point

In preliminary experiments, we found that our objective 3.6 is sensitive to the initialization of the weight in G and F in the purely unsupervised setting. It requires a good initial setting of the parameters to avoid getting stuck in the poor local minimal. To address this sensitivity issue, we employed a similar approach as in [1, 127] to firstly used an adversarial training approach to learn G and F and use them as the initial point for training our full objective 3.6. More specifically, we choose to minimize the earth mover distance (a.k.a optimal transport distance) below.

$$d_{emd}(G) := \min_{P \in U(r,c)} \langle P, M^{(G)} \rangle \tag{3.7}$$

U is the transport polytope without entropy constraint, defined as follows.

$$U = \{P \in {\mathbb{R}_+}^{n \times m} | P\mathbb{1}_m = r, P^T\mathbb{1}_n = c\} \tag{3.8}$$

We optimize the distance above by its dual form and through adversarial training, which is also known as Wasserstein GAN (WGAN) [4]. Instead of using an adversarial classifier like in the generative adversarial network [30], WGAN uses a critic function with a Lipschitz constraint to distinguish two distributions. To enforce this constraint in the optimization, we applied the trick proposed by Gulrajani et al. [36], which has been shown to perform better than standard WGAN.

Although the first phase of WGAN training could be unstable, and the performance is lower than using the Sinkhorn distance, the adversarial training narrows down the search space of model parameters and boosting the training of our proposed model. A typical plot of different training objectives and validation accuracy with regards to the training steps is shown in figure 3.2, where the vertical line represents the switch of training objective from WGAN to Sinkhorn distance. We can see that even the earth mover distance drops quickly after we switch to Sinkhorn distance as objective, and there a significant boost on the validation accuracy. This optimization observation confirms that our model is more stable and effective in closing the distributional gap than WGAN.

Implementation

We implemented transformation G and F by a linear transformation. The dimension of the input and output are the same with the word embedding dimension d.[3] For all the experiments in the subsequent subsection, the β in (3.6) was set to be 0.1. For hyper-parameters from the computation of Sinkhorn distance, we choose $\lambda = 10$ and run the matrix scaling algorithm for 20 iterations.

The input embeddings X and Y are prepossessed to have zero mean and unit variance on each dimension.

In our preliminary experiments, we found that tying the weight matrix of F and G to be the transpose of each other improve the performance. So we keep this configuration in all our experiments. This constraint is well-motivated since $F(G(X))$ could be viewed as an autoencoder where $G(\cdot)$ is the encoder and $F(\cdot)$ is the decoder. And it is a common practice to tie the weights of encoder and decoder in an autoencoder.

We minimized the full objective (6) on mini-batches of size 2048 using RMSprop optimizer [105] at a learning rate of 0.0005. We run WGAN training for the first 2000 epochs and switched to the Sinkhorn objective. We used a learning rate decay of 0.95 if objective fails to decrease in each epoch and early stopped training if the objective stopped to decrease for 2000 epochs.

For input embeddings with large vocabulary size, we found it is not necessary and sometimes harmful to include all the words into our training procedure. Therefore we input the $10,000$ most frequent words in WE-C for each language in our experiments. For smaller WE-Z, we simply use all available given embeddings. As for word frequencies, we simply assume a

[3]We tried more complex non-linear transformations for G and F. The performance is slightly worse than the linear case.

uniform distribution of words, i.e. $r = \mathbb{1}_n/n$, $c = \mathbb{1}_m/m$. We also tried using true word frequencies from the corpus in LEX-Z and task 1[4], but no significant performance improvement was observed.

3.3.2 Empirical Evaluation

We evaluated our approach in comparison with state-of-the-art supervised/unsupervised methods on several evaluation benchmarks for bilingual lexicon induction (Task 1) and word similarity prediction (Task 2). We include our main results in this subsection and report the ablation study in the supplementary material.

Data

Monolingual Word Embedding Data All the methods being evaluated in both tasks take monolingual word embedding in each language as the input data. We use publicly available pre-trained word embeddings trained on Wikipedia articles: (1) a smaller set of word embeddings of dimension 50 trained on comparable Wikipedia dump in five languages [126][5] and (2) a larger set of word embeddings of dimension 300 trained on Wikipedia dump in 294 languages [11][6]. For convenience, we name the two sets **WE-Z** and **WE-C**, respectively. **Bilingual Lexicon Data** We need true translation pairs of words for evaluating methods in bilingual lexicon induction (Task 1). We followed previous studies and prepared two datasets below.

LEX-Z: Zhang et al. [126] constructed the bilingual lexicons from various resources. Since their ground truth word pairs are not released, we followed their procedure, crawled bilingual dictionaries and randomly separated them into the training and testing set of equal size.[7] Note that our proposed method did not utilize the training set. It was only used by supervised baseline methods described in subsection 3.3.2. There are eight language pairs (order counted); the corresponding dataset statistics are summarized in Table 3.1. We use WE-Z embeddings in this dataset.

LEX-C: This lexicon was constructed by Lample et al. [58] and contains more translation pairs than LEX-Z. They divided them into training and testing set. We run our model and the baseline methods on 88 language pairs. For each language pair, the training set contains $5,000$ unique query words and the testing set has $1,500$ query words. We followed Lample et al. [58] and set the search space of candidate translations to be the $200,000$ most frequent words in each target language. We use WE-C embeddings in this dataset.

Bilingual Word Similarity Data

For bilingual word similarity prediction (Task 2) we need true labels for evaluation. Following Lample et al. [58], we used the SemEval 2017 competition dataset, where human annotators measured the cross-lingual similarity of nominal word pairs according to the five-point Likert

[4]LEX-C does not contain word frequency information

[5]Available at `http://nlp.csai.tsinghua.edu.cn/~zm/UBiLexAT`

[6]Available at `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`

[7]The bilingual dictionaries we crawled are submitted as supplementary material.

		# tokens	vocab. size	bi. lex. size
tr-en	tr	6m	7,482	18,404
	en	28m	13,220	27,327
es-en	es	61m	4,774	3,482
	en	95m	6,637	10,772
zh-en	zh	21m	3,349	54,170
	en	53m	5,154	51,375
it-en	it	73m	8,490	4,999
	en	93m	6,597	11,812

Table 3.1: The statistics of LEX-Z. The languages are Spanish (es), French (fr), Chinese (zh), Turkish (tr) and English (en). Number of tokens is the size of training corpus of WE-Z. The bilingual lexicon size means the number of unique words of a language in the gold bilingual lexicons.

scale. This dataset contains word pairs across five languages: English (en), German (de), Spanish (es), Italian (it), and Farsi (fa). Each language pair has about 1,000 word pairs annotated with a real similarity score ranging from 0 to 4.

Baseline Methods

We evaluated the same set of supervised and unsupervised baselines for comparative evaluation in both Task 1 and Task 2. The supervised baselines include the methods of Artetxe et al. [5, 6], Mikolov et al. [71], Shigeto et al. [99], Xing et al. [114], Zhang et al. [131].[8] We fed all the supervised methods with the bilingual dictionaries in the training portions of the LEX-Z and LEX-C datasets, respectively.

For unsupervised baselines we include the methods of Zhang et al. [126] and Lample et al. [58], whose source code is publicly available as provided by the authors.[9]

	Supervised						Unsupervised		
	[71]	[131]	[114]	[99]	[5]	[6]	[58]	[126]	Ours
af-en	20	27.33	26.6	**38.60**	30.07	23.67	0	0	**20.20**
en-af	**33.00**	23.07	23.33	13.47	27.67	22.2	0	0	**17.40**
ar-en	38.42	45.05	45.45	**53.61**	47.05	39.96	0.13	0	**42.37**
en-ar	**40.07**	34.2	34.07	29.6	36.93	26.53	0	0	**25.20**
bg-en	44.8	50.6	50.33	**61.00**	53.27	47.27	26.47	0.47	**49.67**
en-bg	**48.47**	39.73	40	33.8	43.4	34.4	13.87	0	**34.13**
bn-en	14.3	21.46	20.59	**29.68**	22.59	8.29	0	0	**12.50**

[8]The implementations are available from `https://github.com/artetxem/vecmap`.

[9]We used implementation by Zhang et al. [126] from `http://nlp.csai.tsinghua.edu.cn/~zm/UBiLexAT` and that of Lample et al. [58] from `https://github.com/facebookresearch/MUSE`

en-bn	**24.47**	15.6	15.33	7.07	19.87	6.2	0	0	**7.47**
bs-en	22.6	29.67	29.73	**39.93**	31.6	20.8	0	0	**22.73**
en-bs	**31.13**	23.2	23.93	16.33	28.4	13.6	0	0	**12.33**
ca-en	57.73	63.4	63.4	**69.33**	65.27	61.27	41	1	**58.33**
en-ca	**66.20**	58.73	58.53	53.6	60.87	56.73	33.07	0.07	**54.47**
cs-en	50	56.53	56.13	**64.93**	58.67	53.6	**60.00**	0.67	52
en-cs	**54.20**	48.8	48.47	42.53	51.6	40.87	**48.20**	0	37.73
da-en	53.07	58.53	59.07	**68.40**	61.07	57.47	**60.87**	0	55.87
en-da	**58.60**	45.93	45.93	37.27	51.33	45.2	**49.27**	0	44.6
de-en	61.93	67.67	67.73	**71.07**	69.13	68.07	**69.87**	2.13	67
en-de	**73.07**	69.87	69.53	63.73	72.13	69.2	**71.53**	0.93	68.87
el-en	46.67	53.07	52.73	**60.80**	55.27	47.2	**55.73**	0.07	50.33
en-el	**47.00**	39	39	33.33	42.73	35.93	**39.60**	0	34.6
es-en	74	77.27	77.2	**81.07**	78.27	75.6	**78.53**	2.53	77.8
en-es	**80.73**	78.53	78.6	74.53	80.07	78.2	79.4	0.33	**79.53**
et-en	28.2	35.87	37.53	**49.33**	39.67	27.73	**34.53**	0	27.67
en-et	**38.40**	28.4	29.07	23.07	34.6	19.2	**18.33**	0	16.13
fa-en	27.6	33.49	34.03	**40.25**	36.37	26.93	**34.90**	0.2	30.48
en-fa	**39.73**	30.53	30.67	19	34.13	22.33	**28.27**	0	25.6
fi-en	45.93	52.87	53.27	**63.67**	54	43	**57.40**	0.27	47.07
en-fi	45.93	41.47	41.07	37.93	**46.40**	32.33	**38.73**	0	31.73
fr-en	71.33	76.07	76.33	**79.93**	77.73	74.47	**77.67**	1.2	75.33
en-fr	**82.20**	78.2	78.67	73.13	79.2	77.67	**78.33**	0.07	77.93
he-en	41.66	49.27	49	**55.27**	50.4	44.53	**51.47**	1.4	47.2
en-he	**46.87**	37.47	37.13	32.4	42.4	27.6	**36.27**	0	29.93
hi-en	27.06	34.56	35.37	**45.14**	38.65	29.6	0	0	**26.93**
en-hi	**39.07**	27.67	27.67	14.4	33.27	25.2	0	0	**21.20**
hr-en	37.09	44.23	44.3	**54.77**	46.5	35.69	**42.09**	0	40.56
en-hr	**40.67**	33	33.87	27.4	37.27	23.93	**25.67**	0	24.93
hu-en	45.73	54.93	54.8	**65.93**	57.33	47.33	**59.33**	0.07	52.13
en-hu	**56.13**	48.4	47.8	41.4	52.2	40.27	**46.93**	0	39.4
id-en	53.07	58.33	58.8	**66.60**	60.4	56	**60.67**	0	56
en-id	**69.07**	57.2	57.87	43.67	61.2	56.4	**59.00**	0	56.8
it-en	68.93	72.4	72	**76.47**	73.6	70.53	**74.60**	2.27	72.07
en-it	**77.60**	73.4	73.33	68.13	74.47	71.67	**75.80**	0.2	73.87
ja-en	26.19	0	0	29.36	**31.50**	0	0	0	0
en-ja	**55.45**	2.4	2.6	11.31	48.12	0.69	0	0	0
ko-en	16.08	19.3	19.51	**32.44**	24.02	10.4	**18.41**	0	14.17
en-ko	**39.86**	35.97	36.25	25.05	**39.86**	11.95	**19.04**	0	9.01
lt-en	21.07	26.53	26.6	**36.07**	28.93	20.8	0	0	**21.93**
en-lt	**28.13**	23.6	23.4	18	24.73	13.6	0	0	**12.40**
lv-en	26.53	34.53	34.27	**42.20**	35.8	24.07	0	0	**28.13**
en-lv	**28.93**	22.87	21.6	13.87	26.47	12.2	0	0	**12.80**

mk-en	39.4	47.87	47.8	**58.27**	50.4	42.87	**49.67**	0.13	42.93
en-mk	**46.07**	37	36.47	29.4	41.2	30.93	**32.47**	0	28
ms-en	29.62	35.89	36.69	**46.90**	39.03	31.09	0	0	**23.95**
en-ms	**50.47**	39.53	40	24.93	44.33	38.73	0	0.07	**29.60**
nl-en	61.13	65.27	65.53	**72.27**	68.33	64.4	**70.67**	10.2	65.27
en-nl	**74.80**	67.67	66.93	59.53	69.6	66.93	**69.40**	1.67	67.87
no-en	52.2	60.33	60.47	**66.80**	61.6	55.33	**62.20**	0.47	55.73
en-no	**62.47**	53.27	53.33	42.93	57.4	50.93	**55.73**	0.13	50.2
pl-en	53.53	61.47	61.6	**66.13**	63.67	54.6	**62.87**	0.07	58.47
en-pl	**57.80**	52.93	53.6	48.2	55.73	45	**53.33**	0	46.53
pt-en	69.93	73.27	74	**76.93**	74.87	72.2	**75.33**	5.47	72
en-pt	**78.87**	73.73	73.4	66.07	74.8	73.53	**76.07**	0.2	73.53
ro-en	52.93	60.2	61	**69.53**	62.67	57	**63.33**	0	56.4
en-ro	**57.13**	50.2	50	41.53	54.93	47.4	**51.93**	0	44.87
ru-en	50.27	58.2	58.2	**64.07**	60.13	51.2	**58.07**	0	50.93
en-ru	**52.53**	49.4	49.13	46.6	52.47	38.87	**39.33**	0	35.73
sk-en	39.2	48	48.93	**57.27**	51.27	41.4	**48.13**	0	40.87
en-sk	**39.33**	32.93	32.47	26.47	36.8	24.13	**29.33**	0	23.27
sl-en	35.13	42.73	44.07	**53.27**	45.4	35.13	0	0	**37.67**
en-sl	**38.00**	31.6	31.6	24.8	35.33	21.4	0.13	0	**20.07**
sq-en	29.73	35	36.27	**46.33**	38.2	27.8	0	0	**27.73**
en-sq	**27.80**	18.73	19.87	13.07	25.27	13.47	0	0	**12.07**
sv-en	43.73	50.87	51.13	**61.27**	54.07	38.07	24.27	0	**48.13**
en-sv	**63.73**	53.93	53.73	41.67	55.93	44.2	24.47	0	**50.47**
ta-en	11.86	18.49	18.02	**25.85**	20.56	7.17	0	0	**9.38**
en-ta	**21.80**	16.2	15.93	11.47	18.8	6.27	0	0	**7.60**
th-en	5.6	7.77	7.9	**14.38**	8.91	0.54	0	0	**3.65**
en-th	**30.53**	25.47	25.87	16.67	28.07	5.33	0	0	**12.53**
tl-en	10.53	14.87	14.2	**23.67**	15.93	7.27	0	0	**7.47**
en-tl	**23.33**	17.53	17.2	7.13	20.73	8.67	0	0	**5.27**
tr-en	42.96	49.43	49.63	**59.77**	52.84	48.03	**55.70**	0	47.5
en-tr	**51.60**	39.67	39.73	31	45.2	34.93	**41.33**	0	34.33
uk-en	36.33	42.73	42.8	**49.20**	44.2	38.87	**45.53**	0.13	41.6
en-uk	**37.47**	31.93	31.47	28.4	33.73	21.87	**27.27**	0.07	23.13
vi-en	28	40.6	39.33	**53.20**	45.8	0.33	**0.13**	0	0
en-vi	**42.40**	28.33	28.87	12	33.6	0.8	**0.07**	0	0
zh-en	29.93	9.27	9.07	31.33	**33.07**	6.4	0	0	0
en-zh	**50.53**	29.33	30.6	13.07	44.87	5.07	0	0	0
mean	44.25	42.41	42.49	43.31	**46.55**	35.64	31.61	0.37	**35.81**
std	17.08	18.66	18.68	18.08	16.94	21.74	16.30	1.30	22.45
failure times	0	2	2	0	0	5	30	86	6

	Methods	tr-en	en-tr	es-en	en-es	zh-en	en-zh	it-en	en-it
Supervised	Mikolov et al. [71]	19.41	10.81	68.73	41.19	45.88	**45.37**	59.83	**41.26**
	Zhang et al. [131]	23.39	**11.07**	72.36	41.19	48.01	42.66	**63.19**	40.37
	Xing et al. [114]	24.00	10.78	71.92	41.02	48.10	42.90	62.81	40.43
	Shigeto et al. [99]	**26.56**	8.52	72.23	37.80	**49.95**	38.15	63.14	35.63
	Artetxe et al. [5]	23.49	10.74	71.98	41.12	48.01	42.66	63.14	40.28
	Artetxe et al. [6]	22.88	10.78	**72.61**	**41.62**	47.54	42.82	61.32	39.63
Unsupervised	Lample et al. [58]	4.09	1.41	60.16	33.58	41.98	34.70	26.98	15.47
	Zhang et al. [126]	15.83	7.41	63.41	37.73	42.08	41.26	54.75	37.17
	Ours	**23.29**	**9.96**	**73.05**	**41.95**	**49.03**	**44.63**	**61.42**	**39.63**

Table 3.2: The accuracy@1 scores of all methods in bilingual lexicon induction on **LEX-Z**. The best score for each language pair is bold-faced for the supervised and unsupervised categories, respectively. Language pair "A-B" means query words are in language A and the search space of word translations is in language B. Languages are paired among **English(en), Turkish (tr), Spanish (es), Chinese (zh) and Italian (it)**.

Results in Task 1

Bilingual lexicon induction is a task to induce a translation in the target language for each query word in the source language. After the query word and the target-language words are represented in the same embedding space (or after our system maps the query word from the source embedding space to the target embedding space), the k nearest target words are retrieved based on their cosine similarity scores with respect to the query vector. If the k retrieved target words contain any valid translation according to the gold bilingual lexicon, the translation (retrieval) is considered successful. The fraction of the correctly translated source words in the test set is defined as $accuracy@k$, which is a conventional metric in benchmark evaluations.

Table 3.2 shows the accuracy@1 for all the methods on LEX-Z in our evaluation. We can see that our method outperformed the other unsupervised baselines by a large margin on all the eight language pairs. Compared with the supervised methods, our method is still competitive (the best or the second-best scores on four out of eight language pairs), even ours does not require cross-lingual supervision. Also, we notice the performance variance over different language pairs. Our method outperforms all the methods (supervised and unsupervised combined) on the English-Spanish (en-es) pair, perhaps for the reasons that these two languages are most similar to each other, and that the monolingual word embeddings for this pair in the comparable corpus are better aligned than the other language pairs. On the other hand, all the methods including ours have the worst performance on the English-Turkish (en-tr) pair. Nevertheless, the relative performance of our method compared to others is quite robust over different language pairs.

Table 3.3 summarize the results of all the methods on the 88 language pairs of LEX-C dataset. On the bottom of the table, we report the mean and standard deviation of the performances for each method. In addition, we define that a method to be failed on a language pair if the

accuracy@1 is less than 5%. And we count the number of failure cases on the bottom line of table 3.3.

Several points may be worth noticing. Firstly, the performance scores on LEX-C are not necessarily consistent with those on LEX-Z (Table 3.2) even if the methods and the language pairs are the same; this is not surprising as the two datasets differ in query words, word embedding quality, and training-set sizes. Secondly, the performance gap between the best supervised methods and the best unsupervised methods in Table 3.3 are larger than that in Table 3.2. This is attributed to a large amount of good-quality supervision in LEX-C (5,000 human-annotated word pairs) and the larger candidate size in WE-C ($200,000$ candidates). Thirdly, our method performs significantly better than the state-of-art unsupervised baselines on the averaged accuracy@1 and failure times. Comparing with supervised methods, our method is worse than most of them on average, indicating that high-quality bilingual dictionary is crucial for better performance for some language pairs. Lastly, as observed in LEX-Z, the performance also varies across languages, indicating that some language pairs are harder than the others. Combining all these observations, we see that our method is highly robust for various language pairs and under different training conditions.

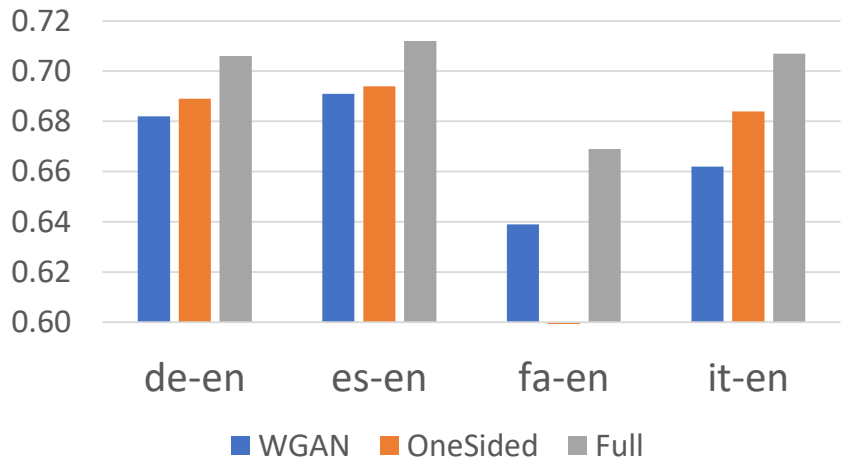


Figure 3.3: Pearson correlation for cross-lingual semantic word similarity task for ablation study

Results in Task 2

We evaluate models on cross-lingual word similarity prediction (Task 2) to measure how much the predicted cross-language word similarities match the ground truth annotated by humans. Following the convention in benchmark evaluations for this task, we compute the Pearson correlation between the model-induced similarity scores and the human-annotated similarity scores over testing word pairs for each language pair. A higher correlation score with the ground truth represents the better quality of induced embeddings. All systems use the cosine similarity between the transformed embedding of each query and the word embedding of its paired translation as the predicted similarity score.

Table 3.4 summarizes the performance of all the methods in cross-lingual word similarity prediction. We can see that the unsupervised methods, including ours, perform equally well as the supervised methods, which is highly encouraging.

	Methods	de-en	es-en	fa-en	it-en
Supervised	Mikolov et al. [71]	0.71	0.72	0.68	0.71
	Zhang et al. [131]	0.71	0.71	0.69	0.71
	Xing et al. [114]	0.72	0.71	0.69	0.72
	Shigeto et al. [99]	0.72	**0.72**	0.69	0.71
	Artetxe et al. [5]	**0.73**	**0.72**	**0.70**	**0.73**
	Artetxe et al. [6]	0.70	0.70	0.67	0.71
Unsupervised	Lample et al. [58]	**0.71**	**0.71**	**0.68**	**0.71**
	Zhang et al. [126]	-	-	-	-
	Ours	**0.71**	**0.71**	0.67	**0.71**

Table 3.4: Performance (measured using Pearson correlation) of all the methods in cross-lingual semantic word similarity prediction on the benchmark data from Lample et al. [58]. The best score in the supervised and unsupervised category is bold-faced, respectively. The languages include English (en), German (de), Spanish (es), Persian (fa) and Italian (it). "-" means that the model failed to converge to reasonable local minimal during the training process.

Ablation Study

We verify the necessity of our system choice by an ablation study. Our first ablated model changes the symmetric objective function to be one-sided. More specifically we let the full objective to be:

$$d_{sh}(G) + \sum_i 1 - \cos(x_i, F(G(x_i)))$$

The similar change is also applied to the initial point searching process with adversarial training. We refer this model as **OneSided**.

Our second ablated model uses only the WGAN training procedure described in subsection 3.4. We found it difficult to find stopping criterion for WGAN based on its objective, therefore we used our full objective (6) to select model during training. We refer this model as **WGAN**. The comparison between the two ablated model and our originally proposed model is shown in figure 3.3.2.

The results with a subset of LEX-C are shown in figure 3.3.2 and figure 3.3 for the task 1 and task 2 respectively.

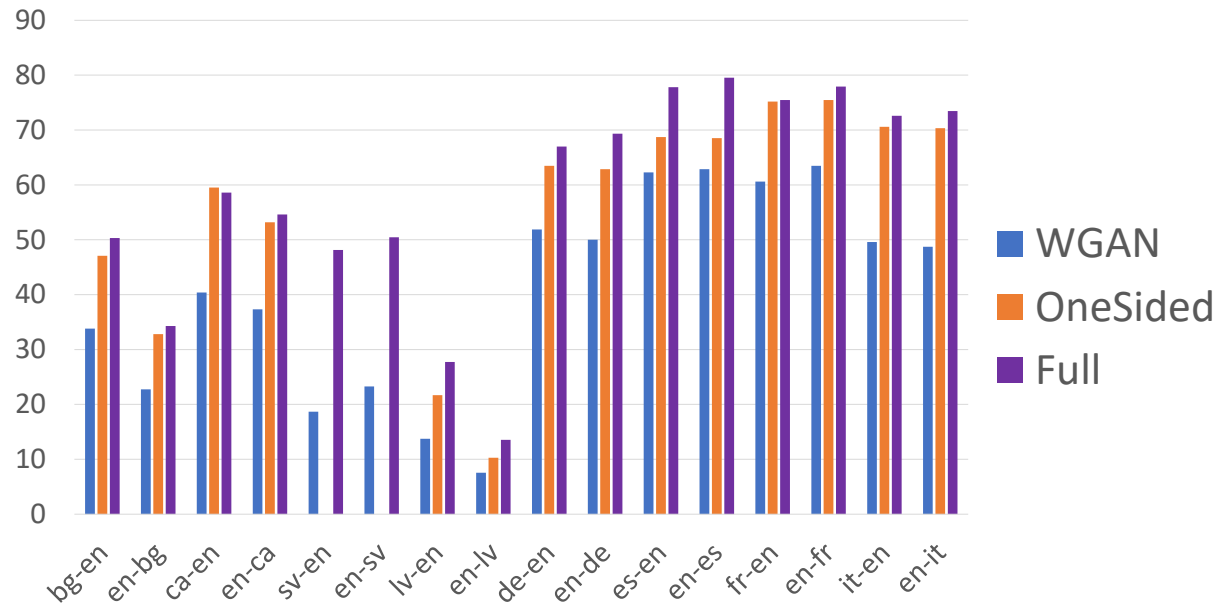


Figure 3.5: Performance of our model and ablated models on LEX-C dataset

.

We can see that WGAN always produces the worst cross-lingual transformation. Although the adversarial training is good at search the parameter space, it is not good at converging to the well-performing local optimal. OneSided is better than WGAN for most of times, but failed to find reasonable initial point sometimes("fa-en" in figure 3.3 and "sv-en", "en-sv" in figure 3.3.2). Overall, our model with full objective achieves the best performance on all tasks compared with ablated models. The only exception is "ca-en" for LEX-C. These results justified our system choice.

	Category	Gold	Prediction
1	Grammatical conjugation (es)	permitir [permit/allow]	permitió (preterite form in the third person singular)
	Articles (bg)	интервю [interview]	интервюто [the interview]
2	Numbers	шестдесет [sixty]	десет [ten]
	Singular/plural	общност [community]	общностите [communities]
	Antonym	inicia [to start]	termina [to end]
3	Person	Reyes [Reyes]	Hernández [Hernandez]
	City	дарбишър [Derbyshire]	пазарджик [Pazardzhik]

Таблица 3.5: Typical errors on the en-es and en-bg translation tasks.

Error Analysis

To gain the insights to further improve our method, we conducted error analysis for en-es and en-bg translation on LEX-C, where our model achieved the higher or on par performance with supervised methods. Although we only used 1,500 query words in our experiments, we consider the 10k most frequent English words[10] as query words to better understand the errors the model tends to make. We randomly sampled 100 failure cases from each language pair and analyzed each case. Our key observations are three-fold (Table 3.5):

1. The evaluation results for both the language pairs contain not a few false negatives (14 for en-es and 23 for en-bg), namely correct translations but not included in LEX-C. Such cases for en-es mostly came from the different forms of verbs because Spanish verbs are rich in grammatical conjugation. The false negatives for en-bg are typically nouns because articles are postfixed to nouns but LEX-C does not cover such forms sufficiently.
2. The model often confuses similar- or opposite-meaning entities because their word embeddings are often very similar. This is a well-known problem of word embeddings in general.
3. The evaluation set contains some proper nouns such as person names and city names, and they are typically very difficult to generate exact translations. However, translating proper nouns is not necessarily important in downstream cross-lingual tasks, and we could ignore them when we evaluate the cross-lingual transfer of word embeddings.

[10]Released by Lample et al. [58]

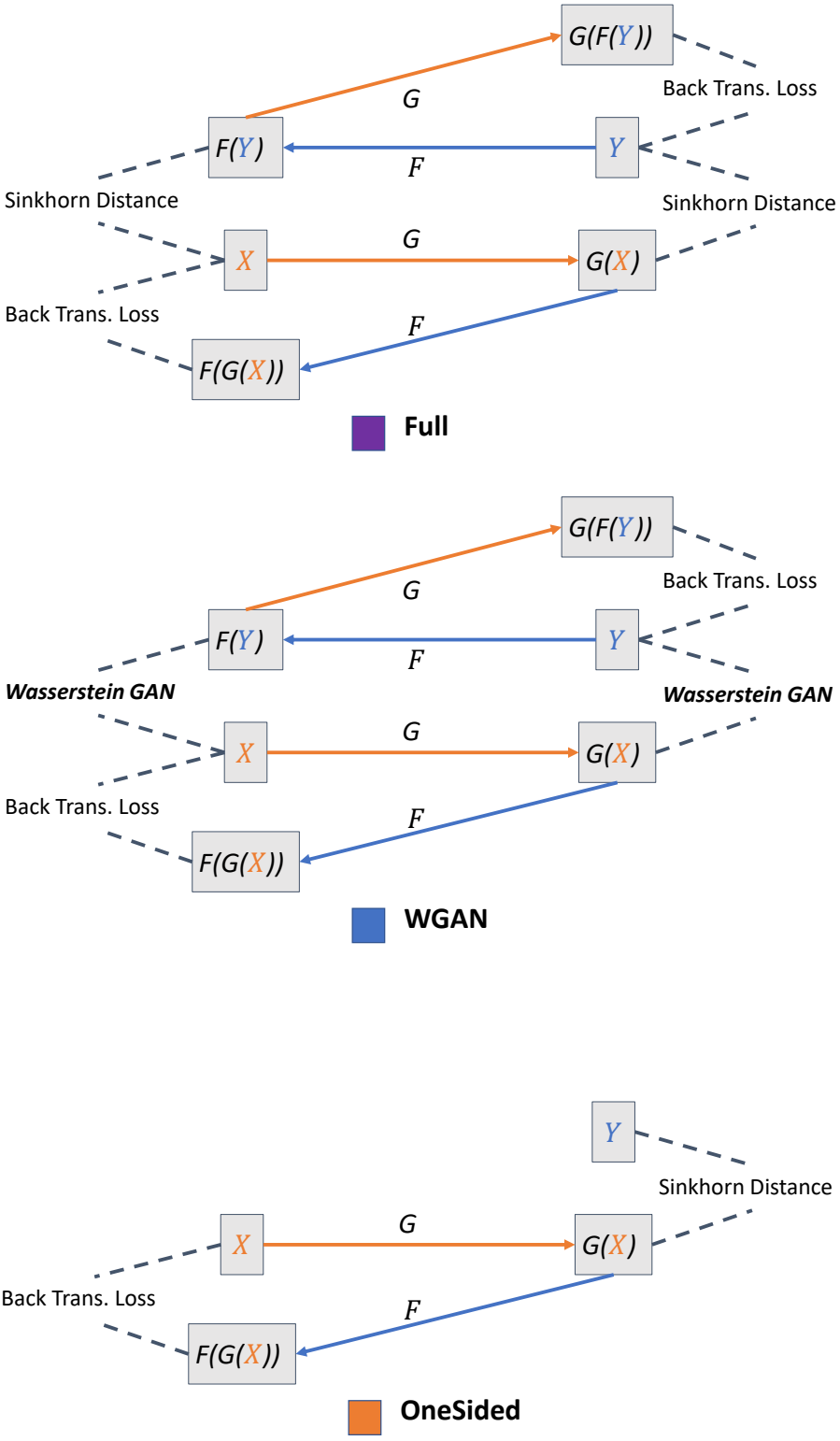


Figure 3.4: Systematic figures to compare our proposed model with two ablated models: WGAN and OneSided

.

Chapter 4

Transfer of Styles and Attributes of Sentences

4.1 Motivation

In chapter 3, we investigated methods to transfer word embedding across language, which could further enable the transfer of neural-based NLP models or word-to-word translations. In the rest of the thesis, we explored the problem of directly transferring the sentence-level text on its style or attribute, where styles broadly include formality, brevity, and attributes include sentiment and topic. For the task of style and attribute transfer (SAT) in the text, one of the most prominent challenges is the lack of parallel corpus which contains paired sentences with different style/attribute. On the other hand, unpaired sentences in different styles or attributes are relatively easy to access. For instance, there are numerous datasets for sentiment classification [107]. Therefore, in this chapter, we are going to propose 1) a semi-supervised sentence transfer model that could be trained on a limited amount of parallel corpus and 2) an unsupervised model that trains solely on attribute-classified sentences and requires no parallel corpus. The effectiveness of our method was verified on the benchmark datasets of formality and sentiment transfer.

4.2 Semi-supervised Transfer

Background

Text SAT is an important research topic in natural language generation since specific styles or attributes of text are preferred in different cases [85, 96]. Early work of SAT relies on parallel corpora where paired sentences have the same content but are in different styles. For example, Xu et al. [119] studied the task of paraphrasing with a target writing style. Trained on human-annotated sentence-aligned parallel corpora, their model can transfer text into William Shakespeare's style.

However, the genre and amount of parallel corpora are very limited for text SAT tasks. Therefore, recent work focuses on eliminating the requirement for parallel corpora [21, 25, 47,

61, 69, 76, 97, 104, 115]. A common strategy is to first learn a style-independent representation of the content in the input sentence. The output sentence is then generated based on the content and the desired style. To this end, some approaches [25, 47, 69, 76, 97] leverage auto-encoder frameworks, where the encoder generates a hidden representation of the input sentence. Other works including [61, 115, 130], on the other hand, explicitly deletes those style-related keywords to form style-independent sentences.

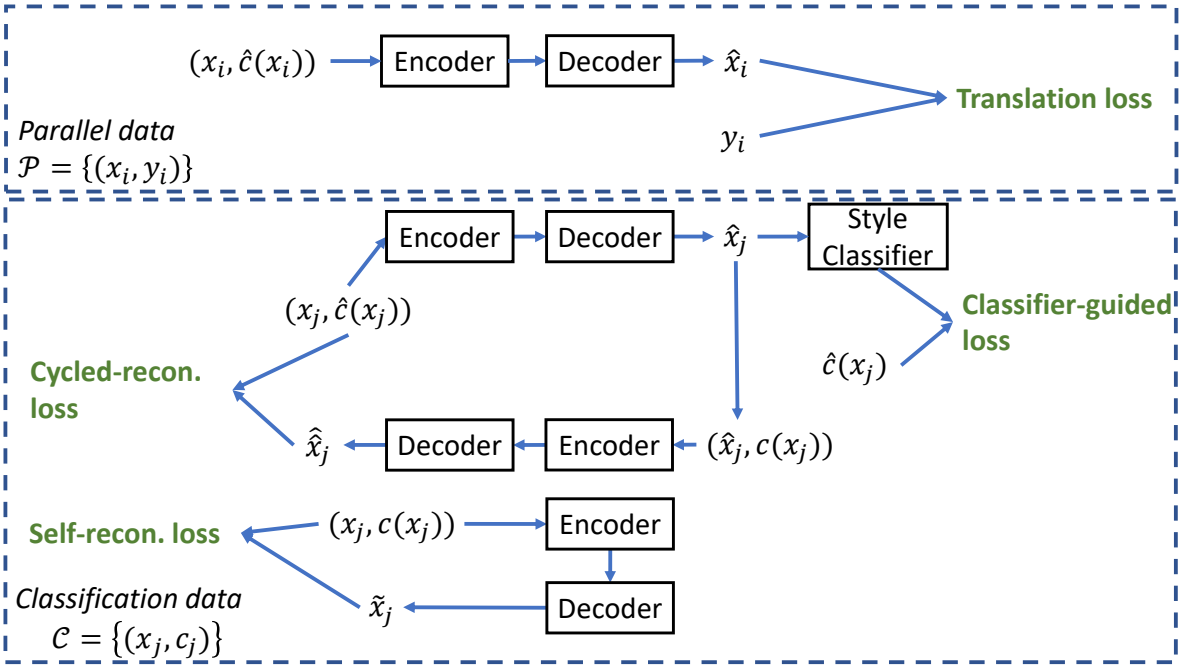


Figure 4.1: The model architecture and various losses for the formality transformer model. All the encoders(decoders) in the figure refer to the same model which appears repeatedly in different loss functions. We use $\tilde{x}_j$ to represent x_j after self-reconstruction and $\hat{\hat{x}}_j$ to represent x_j after cycled-reconstruction

The key insight of these unsupervised methods for text SAT is to disentangle style information from content. To achieve this, an adversarial classifier [25, 97], a collaborative classifier [69], or language models [123] is used to guide the generation process. Some work [69, 115] applies reconstruction losses to ensure that the semantic content of the input sentence can be recovered after style transformation.

We take the insights from these unsupervised methods and propose a novel formality transfer model that is able to incorporate hybrid types of textual annotations. As Figure 4.1 shows, our approach involves a sequence-to-sequence (seq2seq) encoder-decoder model for style transformation and a style classification model. The proposed seq2seq model simultaneously handles the bidirectional transformation of formality between informal and formal, which not only enhances the model's data efficiency but also enables various reconstruction losses to help model training; while the classification model fully utilizes the less expensive formality-classified an-

Informal	I'd say it is punk though.
Formal	However, I do believe it to be punk.
Informal	Gotta see both sides of the story.
Formal	You have to consider both sides of the story.

Table 4.1: Examples from dataset introduced by [88], the formal sentences are the rewrites from the informal ones annotated by human experts.

notation to compute a classifier-guided loss as additional feedback to the seq2seq model.

Our model is also close to semi-supervised machine translation, where recent efforts have focused on incorporating monolingual corpora in addition to the parallel corpora via multi-task learning [35, 125], back-translation [93] or dual learning [42]. Our model borrows the idea of back-translation in MT to enforce the consistency after a cycled transformation of formality (e.g. from informal to formal then from formal back to informal). Different from semi-supervised MT methods, our model also uses a self-reconstruction loss and a classifier-guided loss, which has only been explored in the unsupervised SAT setting.

Experiments on a benchmark formality transfer dataset demonstrate that our approach could achieves the best performance on formality SAT.

To summarize, our major contributions include:

- We advanced the state-of-art on the formality transfer task, with a novel approach that could be trained on both limited parallel data and larger unpaired data.
- We proposed a bi-directional text SAT framework that can transfer formality from formal to informal or from informal to formal with one single encoder-decoder component, enhancing the models' data efficiency. With jointly optimized against various losses, the models can be better trained and yield a promising result.

Transferring the formality is an important dimension of style transfer in text [44], whose goal is to change the style of a given content to make it more formal. The formality transfer system would be useful in addition to any writing assistant tool. To illustrate the desired output of a formality transfer system, we show in table 4.1 the examples of formal rewrites of informal sentences from a recent formality transfer dataset [88].

One typical solution to formality transfer could be the seq2seq encoder-decoder framework [9], which has been successful for machine translation and other text-to-text tasks. Given an informal sentence $\boldsymbol{x} = (x_1, \cdots, x_M)$ and its corresponding formal rewrite sentence $\boldsymbol{y} = (y_1, \cdots, y_N)$ in which x_M and y_N are the M-th and N-th words of sentence $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, the seq2seq model learns a probabilistic mapping $P(\boldsymbol{y}|\boldsymbol{x})$ from the parallel sentence pairs through maximum likelihood estimation (MLE), which learns model parameters $\boldsymbol{\Theta}$ to maximize the following equation:

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{P}} \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\Theta}) \tag{4.1}$$

where $\mathcal{P}$ denotes the set of the parallel informal-formal sentence pairs.

4.2.1 Proposed Method

Joint Training with Hybrid Textual Annotation

Notwithstanding the seq2seq model's effectiveness, it requires large amounts of parallel data for training to update its millions of parameters. Unfortunately, the parallel data for formality SAT is expensive. As a result, there are only a limited number of informal-formal parallel sentence pairs available for training, which hinders training a good seq2seq model with the conventional training method that largely relies on the parallel data. To address the limitation of the conventional seq2seq training, we propose a novel joint training approach that is able to train a seq2seq model jointly from hybrid textual annotations (i.e., from both parallel annotation and class-labeled annotation).

Assume we have two sets of data: $\mathcal{P}$ and $\mathcal{C}$. $\mathcal{P}$ is the set of parallel sentence pairs: $\mathcal{P} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{\|\mathcal{P}\|}$, where the i-th sentence pair contains informal sentence $\boldsymbol{x}^{(i)}$ and its corresponding formal re-writing $\boldsymbol{y}^{(i)}$ and $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ are expected to only differ in terms of their formality of expression while their semantic content must be the same. $\mathcal{C}$ is the classification data: $\mathcal{C} = \{(\boldsymbol{x}^{(j)}, c^{(j)})\}_{j=1}^{\|\mathcal{C}\|}$. $\boldsymbol{x}^{(j)}$ is a sentence with the formality label $c^{(j)}$ where $c^{(j)} \in \{\text{informal}, \text{formal}\}$. Specially, we use $c(\boldsymbol{x})$ to represent the known formality label for a given sentence $\boldsymbol{x}$, and $\hat{c}(\boldsymbol{x})$ to represent to the opposite formality of $c(\boldsymbol{x})$.

We propose a novel approach to fully exploit $\mathcal{P}$ and $\mathcal{C}$ by jointly training a seq2seq style transformation model parameterized by $\boldsymbol{\Theta_{s2s}}$ and a classifier parameterized by $\boldsymbol{\Theta_c}$ to minimize the losses described in the sub-sections below.

Bidirectional Translation Loss

As most text-to-text tasks, the most direct way to train a model is to minimize the translation loss as defined in Eq (4.1). In contrast to the conventional seq2seq model that transfers style in only one direction, we propose to model bidirectional style transformation (i.e., both from informal to formal and from formal to informal) with one single encoder-decoder component. We will show in the following sections that the bidirectional style transformation modeling can not only make full use of the parallel data but also enable various reconstruction constraints to help the models learn better from massive monolingual data, which enhances the models' data efficiency. Different from the conventional seq2seq model where only a source sentence is fed into the model, we also tell the model a direction indicator. A special token "<to_formal>"(or "<to_informal>") is appended at the beginning of the input sentence and fed into the encoder in order to specify the direction of the transfer.

For modeling bidirectional style transformation, for each paired sentence $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{P}$, we minimize the negative log likelihood of generating $\boldsymbol{y}$ given $\boldsymbol{x}$ and the reverse direction.

$$L_{trans}(\boldsymbol{\Theta_{s2s}}) = -\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{P}} \log P(\boldsymbol{y}|\boldsymbol{x}, \hat{c}(\boldsymbol{x}); \boldsymbol{\Theta_{s2s}}) + \log P(\boldsymbol{x}|\boldsymbol{y}, \hat{c}(\boldsymbol{y}); \boldsymbol{\Theta_{s2s}})$$

As shown in Eq (4.2), the translation loss is defined on both directions of a sentence pair in parallel data. With shared parameters for bi-directional translation, the model can be trained

from parallel sentence pairs in both directions, making the size of training data twice and accordingly improving the model's data efficiency.

However, the size of parallel data $\|\mathcal{P}\|$ is still too small to train a well-generalized seq2seq model for formality transfer. To avoid the overfitting problem, we further introduce the following losses.

Classifier-guided Loss

To assist model training from the limited parallel data, we propose to use the classification data $\mathcal{C}$ which is much more easily accessible than the parallel data. We first train a classifier to predict the formality of a given sentence. The objective for training the formality classifier is the standard negative log-likelihood loss given in equation 4.2

$$L_{clas}(\boldsymbol{\Theta_c}) = -\sum_{(\boldsymbol{x},c(x))\in\mathcal{C}} \log P(c(x)|\boldsymbol{x};\boldsymbol{\Theta_c}) \tag{4.2}$$

After the classifier $\boldsymbol{\Theta_c}$ is learned, we keep its parameters fixed and use it to update the seq2seq model. Given an informal sentence $\boldsymbol{x}$ and the desired formality $\hat{c}(x)$ (formal), we let Seq2Seq$(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$ be the transferred sentence given by the seq2seq model($\boldsymbol{\Theta_{s2s}}$). In other words, we assume we can find

$$\text{Seq2Seq}(\boldsymbol{x}, \hat{c}(\boldsymbol{x})) = \arg\max_{y} P(y|\boldsymbol{x}, \hat{c}(\boldsymbol{x}); \boldsymbol{\Theta_{s2s}})$$

Since in classification data $\mathcal{C}$, we do not have the ground truth of Seq2Seq$(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$, we cannot optimize $\boldsymbol{\Theta_{s2s}}$ as in Eq 4.2. Alternatively, we can optimize classification loss given by the trained formality classifier in order to let Seq2Seq$(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$ look like the desired style $\hat{c}(\boldsymbol{x})$. The loss is shown in Eq (4.3):

$$\begin{aligned} &L_{clas-guided}(\boldsymbol{\Theta_{s2s}}) = \\ &-\sum_{(\boldsymbol{x},c(x))\in\mathcal{C}} \log P(\hat{c}(\boldsymbol{x})|\text{Seq2Seq}(\boldsymbol{x}, \hat{c}(\boldsymbol{x})); \boldsymbol{\Theta_c}) \end{aligned} \tag{4.3}$$

Differentiable Decoding

In order to generate Seq2Seq(x, c), the decoder samples the output sequence $y_1, y_2, ..., y_{L_y}$ of tokens one element at a time. The process is auto-regressive in the sense that previously generated tokens are fed into the decoder to generate the next token. In its original formulation, the classifier-guided loss (equation 4.3) contains discrete samples generated in such auto-regressive manner, which hinders the gradient propagation. To solve this, we apply a recent technique [47, 97] to approximate the discrete decoding process with a differentiable one. Instead of feeding a discretely sampled token to the decoder, we feed the softmax distribution over the vocabulary as the generated soft word. Let the output logit vector at time step t be v_t. The output for the decoder is softmax(v_t/τ), where $\tau \in (0, 1)$ is the temperature hyper-parameter to control the shape of the softmax function. For the embedding look-up layer of the decoder and

the classifier, we simply take "soft" word embedding by averaging over the word embedding matrix. The generated soft embedding is differentiable w.r.t. the parameters in the decoder and encoder, which enables the gradient from the classifier to propagate back and update the formality transfer model.

Reconstruction Loss

One potential issue of using classifier-guided loss is that the seq2seq model could easily minimize the classification loss defined in Eq (4.3) by simply generating keywords for $\hat{c}(\boldsymbol{x})$. In this case Seq2Seq$(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$ will be classified to the target formality class but become independent to input $\boldsymbol{x}$. To overcome this problem, we propose two reconstruction losses that are easily introduced based on our bi-directional transformation modeling framework.

The first loss is the self-reconstruction loss, which encourages the seq2seq model to reconstruct the input itself if the desired formality is the same as the input one. The objective is similar to Eq (4.2) and is defined using the maximum likelihood as in Eq (4.4).

$$L_{self-recon}(\boldsymbol{\Theta_{s2s}}) = \\ -\sum_{(\boldsymbol{x},c(x))\in\mathcal{C}} \log P(\boldsymbol{x})|\boldsymbol{x}, c(\boldsymbol{x}); \boldsymbol{\Theta_{s2s}}) \tag{4.4}$$

In other words, the self-reconstruction loss makes the seq2seq model leave the input sentence untouched if the desired formality already exists in the input.

The second reconstruction loss requires the seq2seq model's ability to reconstruct the input sentence after a looped transformation which first transfers the input sentence to the target formality and then transfers the output back to its original formality. Let $\hat{\boldsymbol{x}} = \text{Seq2Seq}(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$, the cycled-reconstruction loss is defined in Eq (4.5):

$$L_{cyc-recon}(\boldsymbol{\Theta_{s2s}}) = -\sum_{(\boldsymbol{x},c(x))\in\mathcal{C}} \log P(\boldsymbol{x}|\hat{\boldsymbol{x}}, c(\boldsymbol{x}); \boldsymbol{\Theta_{s2s}}) \tag{4.5}$$

One-sided Cycle Approximation

The discrete generation also exists in the cycled reconstruction loss in Eq (4.5). Instead of using the differentiable decoding, we take a simpler approach by only back-propagating the gradient until the generation of Seq2Seq$(\boldsymbol{x}, \hat{c}(\boldsymbol{x}))$. In other words, we take $(\text{Seq2Seq}(\boldsymbol{x}, \hat{c}(\boldsymbol{x})), \boldsymbol{x})$ as a pair of pseudo-parallel data. The approximation has the same form of back-translation as used in machine translation [94]. The difference is that our model works in monolingual data and that we have a formality label c to control the direction of transformation, whereas in machine translation the back-translation needs a separate back-translator of the reverse direction.

Overall Objective

The overall objective of our seq2seq model is the weighted summation of the various losses we defined above, except for the classification loss in Eq (4.6), which is defined on formality classifier and is optimized as a separate step.

$$
\begin{aligned}
L_{all}(\boldsymbol{\Theta_{s2s}}) =& w_t L_{trans} + w_c L_{clas-guided} \\
&+ w_{sr} L_{self-recon} + w_{cr} L_{cyc-recon}
\end{aligned} \tag{4.6}
$$

Model Configuration

In this section, we illustrate the detailed configuration of the seq2seq model and the classification model in our approach as well as our post-processing steps for formality SAT.

Formality Transformer Model

For the seq2seq model, we use the recently proposed transformer model [106]. In contrast to conventional RNN seq2seq models, a transformer model applies self-attention to the source sentence, which intuitively benefits disambiguation of word sense in an informal sentence and should accordingly yield a better SAT result. To the best of our knowledge, this is the first attempt to adapt the transformer model on formality transformation.

We implement the transformer model based on open source sequence-to-sequence software Fairseq-py [29] For the transformer model, we use the same configuration for both the encoder and decoder. We set the embedding dimension to 256 and the hidden dimension of the feed-forward sub-layer to 1024. The number of layers is set to 2 and the number of heads is set to 4. For the remaining hyper-parameters such as the dropout rate and the activation function, we followed the default choice of Fairseq-py in our implementation. For temperature τ, we anneal it from 1.0 to 0.1 as training proceeds.

For hyper-parameter tuning, we performed grid search over intervals $[0.1, 0, 2, 0.5, 1.0]$ for each weight in equation 4.6 in the development set of GYAFC dataset(see section 4.2.2).

Formality Classification Model

For our formality classification model, we use a CNN text classifier [52]. Given an input sentence x, we first embedded each token with word embedding layer. Then convolutional filters of size n is applied on the embedded sentence, which acts as n-gram feature extractors on the different position of the input sentence. We then take the maximum of the extracted features over different positions with a max-pooling layer. The final feed-forward layer has softmax activation to produce the probability of a given formality.

For implementation details, we use filter size $n = \{3, 4, 5\}$ and filter number 100 for each size. The dropout rate is set to 0.5.

Post-processing

Classifier-based Filtering We filter some of the n-best sentences with our formality classifier. The sentences with the incorrect formality class given by the classifier are removed from the candidate list. And the remaining one with the highest generation score is selected as the final output.

Grammatical Error Correction Motivated by the fact that a formal sentence should be grammatically correct, we feed the output from our formality transfer system to a grammatical error correction (GEC) model as post-processing to get rid of the grammatical mistakes. Specifically, we used the state-of-the-art GEC system [28] which is based on a convolutional seq2seq model with special training and inference mechanism to correct grammatical errors in target sentences.

4.2.2 Empirical Evaluation

Experimental Setting

We used Grammarly's Yahoo Answers Formality Corpus (GYAFC) [88] as our evaluation dataset. It contains paired informal and formal sentences annotated by humans. The dataset is crawled and annotated from two domains in Yahoo Answers [1], namely Entertainment & Music (E&M) and Family & Relationships (F&R) categories. Following the analysis in [88], we only consider the transformation from informal to formal. The statistics of the training, validation and testing splits of GYAFC dataset is shown in table 4.2.

	Train	Validate	Test
E&M	52595	2877	1416
F&R	51967	2788	1332

Table 4.2: The statistics of train, validate and test set of GYAFC.

All splits in GYAFC are given in the form of parallel data. To obtain the classification data, we apply a simple data extension heuristic. We first train a CNN formality classifier on the training split of the parallel corpora, and then make predictions on the unlabeled corpus of Yahoo Answers. The predictions of either formal or informal with a confidence higher than 99.5% are selected as the classification dataset $\mathcal{C}$, which contains about 1100k formal sentences and 350k informal sentences. During training, our model combines the training data from E&M and F&R. The relative weights for various losses are tuned on the validation set.

Baselines

We compare to the following baseline approaches:

- **Transformer** is the original transformer model [106] that shares the same configurations of our model.
- **Transformer-Combine** is **Transformer** that combines the training data from E&M and F&R. It also uses the outputs from PBMT and back-translation [94] to utilize large amount of unpaired sentences.
- **SimpleCopy** is to simply copy the input sentence as the prediction without any modification.

[1] `https://answers.yahoo.com`

- **RuleBased** is the rule-based method that uses a set of rules to automatically make an informal sentence more formal.
- **PBMT** is a phrase-based machine translation model trained on parallel training data and outputs from **RuleBased**. It also uses self-training on unpaired sentences to boost its performance.
- **NMT** is the encoder-decoder model with attention mechanism [9].
- **NMT-Copy** adds the copy mechanism [33] to **NMT**.
- **NMT-PBMT** is a semi-supervised method that further incorporates outputs from **PBMT** with back-translation. Both domain knowledge and additional unlabeled data were used to train this strong baseline.
- **MultiTask** [78] jointly trains the model on GYAFC and a formality-sensitive machine translation task with additional bilingual supervision. It also uses ensemble decoding, while our model and other baselines all use single model decoding. Therefore the performance is not comparable with other models.

Note that in addition to the first three baselines that we implement by ourselves, the outputs of the remaining baselines are from previous works [78, 88]. The comparison between baselines and our method concerning the components of training objectives is shown in table 4.3.

	Translation loss	Cycled-recon. loss	Classifier-guided loss	Self-recon. loss	Self training
RuleBased					
PBMT	✓				✓
NMT	✓				
NMT-Copy	✓				
NMT-Combine	✓	✓			
SimpleCopy					
Transformer	✓				
Transformer-Combine	✓				
MultiTask	✓				
Ours	✓	✓	✓	✓	

Table 4.3: Baselines with their components of training objectives.

Evaluation Metric

We followed the evaluation metric used in [61, 88] to use BLEU [79] to measure the closeness between the system prediction and the human annotation. Moreover, we use GLEU score [77] as an alternative to BLEU. GLEU was originally introduced in the task of grammatical error correction (GEC), which is generalized and modified from BLEU to address monolingual text rewriting evaluation and shows better correlation with human evaluations than BLEU.

We compare the BLUE and GLEU score for the ground truth annotation of human and **SimpleCopy** on GYAFC dataset in table 4.4. We can see that the relative performance gain between human and **SimpleCopy** with regard to BLEU is much smaller than the one according to GLEU. This indicates that BLEU focuses more about the content preserving, but is not sensitive on the

modified part of text rewriting task. GLEU, on the other hand, is better at measuring the correctness of the modifications made by the system.

	E&M		F&R	
	BLEU	GLEU	BLEU	GLEU
SimpleCopy	45.93	7.39	47.41	6.81
Human	49.99	25.02	51.09	24.84

Table 4.4: BLEU v.s. GLEU: For human performance, we take one human reference as predictions to compare against the remaining three human references. The process is iterated over all the four references and the scores are averaged. **SimpleCopy** is evaluated in the same way. Note that the numbers in this table are not comparable with the ones in table 4.5 where all the four references are used as ground truth.

Results and Analysis

The results for formality transfer on GYAFC dataset is shown in table 4.5. **Ours** represents our approach's results without post-processing, while **Ours w/ class-filter** and **Ours w/ gec** are the results with the post-processing steps introduced in Section 4.2.1.

According to Table 4.5, there are several noteworthy points: Firstly, it is observed that the best single-model performance is **Ours w/ gec**, which outperformed all baseline methods in terms of both BLEU and GLEU in E&M and F&R. Compared to the strongest baselines PBMT and NMT-PBMT, both of which incorporate heuristic rules with the training data, our models are end-to-end neural networks without any prior knowledge of the task. The performance of our single model is also competitive with the state-of-art ensemble model(MultiTask), which also utilizes additional supervision.

Secondly, our model **Ours** outperformed Transformer-Combine by a large margin for both BLEU and GLEU and on both domains. We can conclude that our joint training with hybrid annotation (parallel data and classification data) could significantly improve the performance of the state-of-art seq2seq model on the formality transformation task.

Thirdly, we notice that using classifier-filtered decoding (**Ours w/ class-filter**) outperformed using beam search (**Ours**) with regard to GLEU for both domains and with regard to BLEU for F&R. The relative improvement showed that the formality classifier is helpful not only during the training of seq2seq model but also in the testing phase. Also, we verified that the usage of the GEC system as a post-processing step could further improve the performance of the informal to formal transformation.

Lastly, comparing with **Transformer** and **Transformer-Combine**, we can see that using the training data from both domains could further improve the performance for both BLEU and GLEU, indicating that with a limited parallel corpus, additional annotation from a different domain could still help with the sequence-to-sequence learning.

	E&M		F&R	
	BLEU	GLEU	BLEU	GLEU
RuleBased	60.37	16.48	66.4	18.79
PBMT	66.88	24.38	72.4	26.96
NMT	58.27	22.87	68.26	26.3
NMT-Copy	58.67	22.93	68.09	26.05
NMT-Combine	67.51	24.05	73.78	26.74
SimpleCopy	50.28	7.42	51.66	6.8
Transformer	61.86	21.61	66.69	24.94
Transformer-Combine	65.5	23.94	70.63	25.88
MultiTask*	72.01	25.92	75.35	27.15
Ablt. w/o self-recon	64.53	22.81	70.43	22.92
Ablt. w/o cyc-recon	66.39	23.53	71.71	25.50
Ablt. w/o class-guided	67.90	24.13	72.00	24.64
Ours	69.08	24.37	72.90	24.78
Ours w/ class-filter	68.71	24.64	73.16	25.73
Ours w/ gec	**69.63**	**25.78**	**74.43**	**27.35**

Table 4.5: BLEU and GLEU scores on GYAFC dataset. The dataset has two domains: Entertainment & Music (E&M) and Family & Relationship (F&R). The best single model score under each metric is marked bold. **MultiTask*** is not comparable to other models in the table since it uses more supervised data and ensemble decoding.

Ablation Study

In table 4.5, we include the performance of ablated models (**Ablt. w/o ...**) which exclude one specific loss in the overall objective function. From the results of our ablated models, it is clear that all the losses defined on classification data contribute to the improvement. Another observation is that the performance of **Ablt. w/o self-recon** is even lower than **Transformer-Combine**, which demonstrates the importance of the self-reconstruction loss to prevent classifier-guided loss from drastically changing the content of the input sentence.

Case Study

We sampled some outputs of our model (**Ours w/ gec**) and two strong baselines in the test set and qualitatively compare them with the ground truth.

Case 1	Fail to recognize named entities
source	secondly id pick yellow card then simple plan then bowling for soup and finally red hot chilli peppers .
Ours w/ gec	I would pick a yellow card , then try bowling for soup and finally red hot chilli peppers .
NMT-Combined	Secondly , I would pick yellow card , then Simple Plan , bowling for soup and finally Red Hot Chili Peppers .
PBMT	Secondly , I would choose yellow card , Simple Plan , bowling for soup and finally Red Hot Chili Peppers .
ground truth	I would pick Yellow Card second , then Simple Plan , Bowling for Soup , and finally , Red Hot Chili Peppers .
Case 2	Only transfer part of the input
source	I think its a bit cheezy to name it though , but maybe I maybe name it Jet
Ours w/ gec	I think it is a bit cheezy to name it .
NMT-Combined	I think it is a bit cheezy to name it , but maybe I name it Jet .
PBMT	I think it is a bit cheesy to name it , but maybe I could name it Jet .
ground truth	It 's a cheesy name but you could name it Jet .
Case 3	Fail to transfer informal words and phrases
source	Not at all , I really dont think in Spanish either ... they rap about stupid stuff
Ours	Not at all . I do not think in Spanish either . They rap about stupid stuff .
NMT-Combined	Not at all , I do not think in Spanish either they rap about stupid things .
PBMT	Not at all , I do not think in Spanish either they rap about stupid things .
ground truth	Not at all . I really do not think it is so in Spanish , either - they rap about ridiculous things .

Table 4.6: Some typical failure cases for **Ours w/ gec** on E&M of GYAFC

The most common error case is the failure to handle named entities. (case 1 in table 4.6) Our model fails to recognize them and convert them into upper case. This could also explain why the performance of Entertainment & Music is lower than Family & Relationship, since the former domain contains more named entities.

Incomplete transfer (case 2) and neglect of informal words (case 3) are also frequent among failure cases. Similar to the observation in [88], we notice that our model produced similar errors more often when the input sentence is very informal, not fluent, grammatically incorrect or contains unknown/misspelled tokens.

Case 1	Common informal word replacement
source	im going to see them in may and i cant freakin wait ! ! ! ! ! ! ! ! !
Ours w/ gec	I am going to see them in May and I can not wait .
NMT-Combined	I am going to see them in May and I cannot really wait .
PBMT	I am going to see them in May and I am very wait .
ground truth	I am going to see them in May and I can not wait .
Case 2	Uncommon informal word replacement
source	I LOOOOOVVVVVVVEEE this song SOOO Much ! ! ! ! ! !
Ours w/ gec	I love this song very much .
NMT-Combined	I loovvvvvvveee this song so Much .
PBMT	I loovvvvvvveee this song very Much .
ground truth	I love this song very much .
Case 3	Capitalization
source	and how good of a president do u think he would be ?
Ours w/ gec	How good of a president do you think he would be ?
NMT-Combined	And how good of a president do you think he would be ?
PBMT	How good of a president , do you think he would be ?
ground truth	How good of a president do you think he would be ?
Case 4	Punctuation
source	=) who do u thinks better ?
Ours w/ gec	Who do you think is better ?
NMT-Combined	=) who do you think better ?
PBMT	who do you think is better ?
ground truth	Who do you think is better ?

Table 4.7: Some typical successful cases for **Ours w/ gec** on E&M of GYAFC

On the other hand, we also analyzed the most accurate predictions of our model. Some of the most common cases are summarized in table 4.7. We notice that our model is particularly good at handling common and typical informal words (case 1) such as "Wat" (What) and "2" (to) as well as uncommon informal words (case 2) such as "SOOO" (so). Our model is also able to properly capitalize the first letter of the sentence (case 3) and correctly use the punctuation (case 4). The baseline methods, on the other hand, cannot handle informal word replacement as accurately as our proposed method. The capitalization and punctuation are less challenging and could be processed correctly by both the baseline methods and ours.

Summary

In this section, we present an approach to training formality transfer models from hybrid textual annotations. Based on a bidirectional style transformation seq2seq model, we fully exploit formality style classification data through classification feedback and various reconstruction constraints to assist the model learning. Our approach effectively improved the performance of the base model and achieved new state-of-art results on formality transfer task.

4.3 Unsupervised Transfer

4.3.1 Proposed Method

Motivation

In section 4.2, we have proposed a novel semi-supervised model for SAT and showed its effectiveness on a benchmark formality transfer task with the limited parallel corpus. However, even limited parallel corpus is often not available, posing a notable obstacle for the decent performance of SAT on various styles and attributes. Therefore, we propose to improve over the unsupervised SAT models, which require zero parallel corpora. Recent efforts in unsupervised SAT lead to multiple methods that only require unpaired data to train sequence-to-sequence (Seq2Seq) models. The overall intuition of these methods is to disentangle the attribute from the content. For example, some recent effort tried to separate them in the encoded sentence representation with adversarial classifiers [25, 97], collaborative classifiers [47, 69, 76, 104], language models [104, 123], denoised auto-encoding [59] or pre-trained machine translation systems [83]. However, when applied to SAT, the performance of those methods often suffers from imperfect disengagement of hidden representations [115]. Another family of methods aims to shift the burden by explicitly removing attribute keywords from input text using pre-trained classifiers [61, 115, 130] or statistical machine translation [132]. The performance of such two-step systems, however, heavily relies on the accuracy of the attribute keyword removal step. For instance, if the "pizza" (instead of "delicious") in "pizza is delicious." is incorrectly removed for sentiment transfer, it could be difficult for the following generation model to recover it.

In this section, we propose an unsupervised method for attribute transfer that combines the advantages of both worlds. On one hand, our model avoids the risk of information loss by implicitly separating the attribute and content. To achieve this, our model learns to reconstruct a corrupted version of the input sentence conditioned on its original attribute. We further improve our model with a back-translation loss that encourages the generation to be consistent when transferring attribute back-and-forth.

On the other hand, instead of removing keywords, our model retrieves sentences that are semantically similar to the source input sentence but in the target attribute. We call such sentences as prototypes since comparing the prototypes with the input sentence could indicate proper editing for a good transfer of attribute. An intuitive tuple of (input, prototype, output) is shown in figure 4.2. Unlike previous models that only use the input sentence, our novel attribute editor model takes both the input and the prototype sentence into consideration for more effective and robust SAT.

Our proposed model also closely related to the recent effort of retrieval-based text generation, which has been focused on unconditional language modeling [39], machine translation [34], source code generation [41, 124] and so on. Among all the related works, the most similar one to our proposed method is the work by Li et al. [61], which also uses a search engine to provide evidence for the proper editing of SAT. However, their method removes the attribute keywords in the input sentence before sending it to a search engine, which potentially leads to error propagation. Also, our method achieved better performance by optimizing towards a back-translation loss in addition to the self-reconstruction loss used in [61].

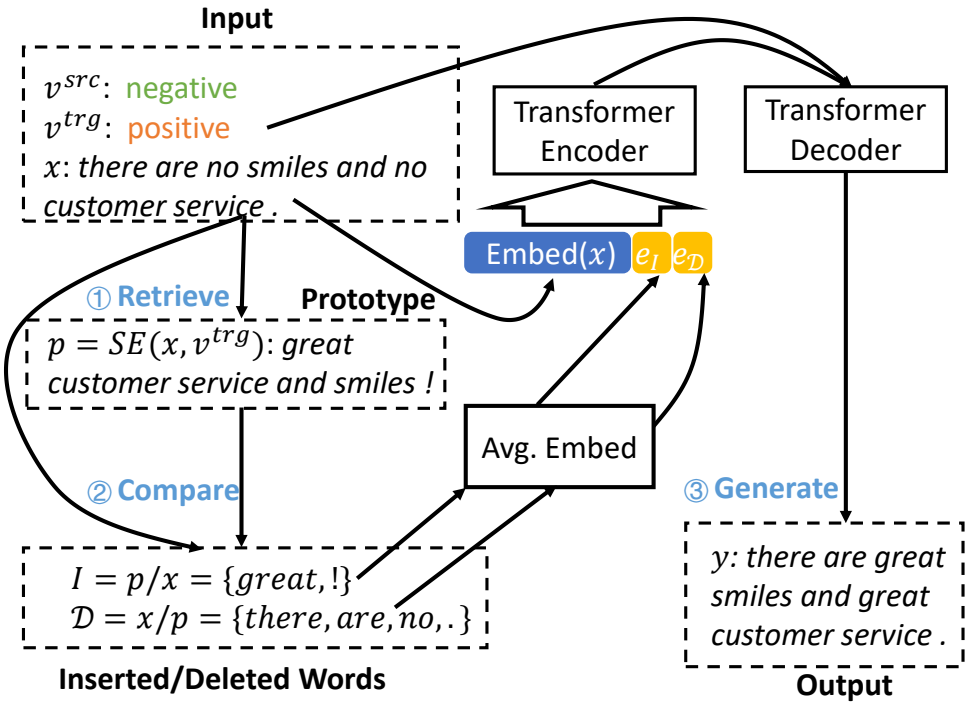


Figure 4.2: System architecture of our proposed model.

Proposed Method

Notations

Our model is trained on a set of labeled sentences $\{(x_1, v_1), ..., (x_m, v_m)\}$, where x_i is a sentence and v_i is the style/attribute for x_i. For instance, $v_i \in$ {"positive", "negative"} for a sentiment transfer problem. At testing time, our model takes a pair of (x, v^{trg}), where x is the sentence whose original attribute is v^{src}, and outputs a sentence that preserves the content of x but holds the target attribute v^{trg}. D_v denotes the collection of sentences that belong to attribute v. And SE(x, v) denotes the most relevant sentence returned by a search engine, where the query is x and the search space is D_v.

Model Overview

The SAT in our model is a three-step process, as illustrated in figure 4.2. Firstly, the search engine uses input x as query and retrieves a prototype from the collection of the target attribute v^{trg} and return $p =$ SE(x, v^{trg}). Secondly, the prototype p is compared with the input x, and the difference is summarized as the inserted/deleted words that convert the input sentence to the prototype. Thirdly, an encoder jointly encodes the input x and the inserted/deleted words. A decoder generates the final output based on the encoded representation and the target attribute v^{trg}.

Model Architecture

For the retrieval step, we choose the BM25 model [92] as our implementation. In our experiment, we use the off-the-shelf implementation from Whoosh [2] as our search engine module.

For sequence generation, we adapt the transformer [106] model, which is widely used as the state-of-art model for the seq2seq generation. In our case, the goal is to allow the generation based on both the input x and the prototype p. To achieve this, we introduce edit vectors $e_{\mathcal{I}}$ and $e_{\mathcal{D}}$ which encodes the difference between the input x and p [39]. For instance, if x and p only differ in a single-word replacement (e.g., replacing "bad" with "good"), then we would expect that the embeddings of the two words capture the semantic difference between the two sentences. Generalizing this intuition to the multi-word difference, we use the averaged word embeddings of the inserted and deleted words. Formally, let $\mathcal{I} = p/\ x$ be the words in p but not in x and $\mathcal{D} = x/\ p$ be the opposite, we have:

$$
\begin{aligned}
e_{\mathcal{I}} &= \frac{\sum_{w \in \mathcal{I}} \text{embed}(w)}{\|\mathcal{I}\|} \\
e_{\mathcal{D}} &= \frac{\sum_{w \in \mathcal{D}} \text{embed}(w)}{\|\mathcal{D}\|}
\end{aligned}
\tag{4.7}
$$

where $\text{embed}(\cdot)$ denotes the embedding look-up layer and $\|\cdot\|$ denotes the size of set.

As illustrated in figure 4.2, $e_{\mathcal{I}}$ and $e_{\mathcal{D}}$ are appended at the end of the embedded input sentence for the transformer encoder. In principle, the self-attention mechanism of encoder should capture the correspondence between the input x and its difference between the prototype, which is summarized in $e_{\mathcal{I}}$ and $e_{\mathcal{D}}$.

Since the generation is supposed to be attribute-dependent, it should also take the target attribute label as input. To achieve this, we use multi-decoder architecture from Fu et al. [25]. We have a single encoder and multiple decoders, each of which is responsible for the generation of one specific attribute. The decoders have the same architecture as the original transformer [106] and use the extracted representation from the encoder to generate output in an auto-regressive left-to-right manner.

Model Objectives

Recall that given x and its source attribute v^{src}, we do not have access to the ground truth outputs that exhibit the target attribute v^{trg}. Instead, we train our model to reconstruct a sentence from its noised version, given its original *source* attribute and the prototype retrieved from the collection of the *source* attribute. To perform well on this loss, the model learns to utilize the prototype for the generation. Furthermore, since we use a separate decoder for each attribute, each decoder learns to generate in its corresponding attribute. Formally, the loss is given by:

$$
L_{recon}(\Theta) = \sum_{x \in X} P(x|\hat{x}, \text{SE}(x, v^{src}), v^{src}; \Theta) \tag{4.8}
$$

[2] `https://whoosh.readthedocs.io`

where $\hat{x}$ is x with some random noise. We introduce noise by randomly deleting or inserting words and permute the order of words in x. The rate of inserting and deleting are both set to 0.15. For permutation, we constrain the permutation within a context window of size 3 [57].

The back-translation loss enforces our model to be consistent when translating back and forth between two attributes. Suppose $\tilde{x}$ is the output from the current model that transfers the attribute of x from v^{src} to v^{trg} and $\tilde{X}$ is the set of all $\tilde{x}$. The back translation loss enforces our model to transfer $\tilde{x}$ back to x when transferring from attribute v^{trg} to v^{src}.

$$L_{bkts}(\Theta) = \sum_{\tilde{x} \in \tilde{X}} P(x|\tilde{x}, \text{SE}(\tilde{x}, v^{src}), v^{src}; \Theta) \tag{4.9}$$

Computing loss l_{bkts} involves searching new queries $\text{SE}(\tilde{x}, v^{src})$ at running time. To speed up, we use the pre-computed $\text{SE}(x, v^{src})$ as an approximation.

The overall objective for our model is the weighted sum of l_{recon} and l_{bkts}.

4.3.2 Emperical Evaluation

Experiment

Dataset

Source	Attribute	Train	Validate	Test
Yelp	Positive	270K	2000	500
	Negative	180K	2000	500
Amazon	Positive	277K	985	500
	Negative	278K	1015	500
ImageCaption	Romantic	6000	300	0
	Humorous	6000	6000	0
	Factual	0	0	300

Table 4.8: The statistics of sentiment transfer datasets from Yelp and Amazon

We tested our model on the Yelp, Amazon and Captions dataset, which are widely used by previous works [61, 83, 97]).

The statistics of the three datasets are shown in table 4.8. The goal for Yelp and Amazon is to transfer the sentiment of an online review either from negative to positive or vice-versa, while the goal of ImageCaption is to transfer between romantic and humorous image captions without the image. Note that ImageCaption has only textual testing sentences, which are supposed to contain no style/attribute, as testing data. We follow the procedure in [61] and treat them as the source sentences without any additional post-processing.

For the ground truth of each test sentence, we use the 5 human references from the work of Jin et al. [49] for Yelp and single human annotation from the work of Li et al. [61] for Amazon and Captions. We used the same train, development and test split as [61].

	Yelp				Amazon				Captions			
	AttrbAcc	BLEU	G-score	GLEU	AttrbAcc	BLEU	G-score	GLEU	AttrbAcc	BLEU	G-score	GLEU
CrossAligned	74.70	19.19	37.86	7.93	74.30	1.81	11.60	1.54	74.10	1.82	11.61	1.57
StyleEmbedding	8.40	42.82	18.97	9.20	54.70	14.60	28.26	8.17	43.30	8.76	19.48	5.93
MultiDecoder	48.30	28.65	37.20	8.29	68.50	8.72	24.44	5.40	68.30	6.63	21.28	4.56
DeleteOnly	86.00	29.97	50.77	10.99	83.00	-	-	-	45.60	11.91	23.30	7.81
DeleteAndRetrieve	89.90	32.43	53.99	11.99	**96.80**	29.59	**53.52**	17.38	48.00	11.94	23.94	7.86
ProtoOnly	**98.10**	12.31	34.75	5.11	94.60	4.61	20.88	3.77	**80.17**	5.19	20.40	3.99
AttributeEditor w/o BT	58.90	**53.02**	55.88	15.58	38.10	37.35	37.72	20.49	61.00	15.73	30.98	10.09
AttributeEditor(Ours)	89.10	51.59	**67.80**	**18.98**	52.90	**37.09**	44.30	**21.78**	61.33	**15.97**	**31.30**	**10.28**

Table 4.9: Automatic metrics w.r.t system output human references in Yelp, Amazon and Captions dataset. "AttrbAcc" stands for accuracy. The ”-” in the table is because the public outputs misaligned with the ground truth.

Baselines

We compared our model with the following groups of baselines:

- **CrossAligned** [97] uses adversarial training to untied attribute from content. The adversarial classifier uses the hidden states of the decoder.
- **StyleEmbedding** and **MultiDecoder** [25], on the other hand, use adversarial classifier on the extracted feature by the encoder. **StyleEmbedding** uses a style/attribute embedding to control the decoding process while **MultiDecoder** uses a separate decoder for each style/attribute.
- **DeleteOnly** and **DeleteAndRetrieve** [61] first remove attribute keywords to get the neutral content. Then the content is modified by the seq2seq model to exhibit the target attribute. Compared with **DeleteOnly**, **DeleteAndRetrieve** further exploit the retrieved keywords from target attribute. Our method differs from them in the way that we do not remove any words from input and the prototype is given as whole sentence instead of keywords.
- **ProtoOnly** directly outputs the prototype sentence retrieved by our search engine.

Implementation

The denoising reconstruction loss and back-translation loss was linearly combined with equal weights as the overall objective.

We processed all sentences into sub-word units [95], which yielded 8123, 15522 6658 unique sub-words for Yelp, Amazon and Captions dataset respectively.

For the transformer model, we used a stack of 2 identical layers for both encoder and decoder. The dimension of input for self-attention was 256, and the dimension for the position-wise feed-forward network was 1024. The number of multi-head attentions was 4. The dropout rates at the input layer, RELU layer, and attention layer were all set to 0.1. The beam size at decoding was set to 12. [3]

[3]The implementation of our sentiment editor model was based on fairseq `https://github.com/pytorch/fairseq`

For optimization, we followed the original transformer work [106] and used Adam optimizer [53] with $\beta_1 = 0.9$, $\beta_2 = 0.997$ and inverse-square-root delay of the learning rate. The maximum learning rate was 0.008, and the number of warm-up steps was 8000. The model was trained for 40 epochs, and the best-performing model on the development split was selected for evaluation.

For search engine, SE(x, v^{src}) should return x as the most relevant sentence because x itself is in the search space. Having exact x as prototype will lead to the trivial solution for our reconstruction loss. Therefore, we instead chose the second most relevant sentence, which is similar but not identical to x.

Greedy decoding is used to generate $\tilde{X}$ on-the-fly for each mini-batch. Since the decoding is discrete, we do not back-propagate the gradient for this generation process. For final decoding on the test set, we used beam search with a size of 12.

Metrics

Following previous works [61, 115], we use BLEU score [79] between the output and the human references to quantify the extent of content preserving. We also use a attribute classifier to measure whether the transferred sentence contains the correct target attribute. The AttrbAcc refers to the fraction of the outputs sentences classified as the target attribute. Due to the commonly observed trade-off between BLEU and AttrbAcc, we follow [115] and compute the harmonic mean of the two, which is known as the G-score. We also included GLEU which generalizes BLEU for monolingual re-writing task [77].

Apart from general metrics on text generation, we are also interested in whether our model truly utilizes the prototype and to which extent the use of prototype leads to the better generation. By comparing the system output h with the input sentence x, we computed the inserted words as $\hat{\mathcal{I}} = h/x$. We define insert ratio (InsR) as $\frac{\|\hat{\mathcal{I}} \cap \mathcal{I}\|}{\|\hat{\mathcal{I}}\|}$, which is the fraction of inserted words in the system output that come from the prototype. The inserted accuracy (InsAcc) is defined as the fraction of words in $\hat{\mathcal{I}}$ that is correct according to any of the human references. Similarly, we also define delete ratio (DelR) and delete accuracy (DelAcc). [4]

Results Analysis

Quantitative The performance of our model (AttributeEditor) and other baseline methods is shown in table 4.9. We notice that our method outperforms all baselines regarding the G-score and GLEU on Yelp and Captions dataset. On amazon dataset, our model tops on GLEU and ranks second on G-score. The strong performance indicates that our model achieves a significantly improved and balanced performance of modifying the style/attribute and keeping the content consistent.

For ablation study, we present the results of the ablated model without the back-translation loss in the second last row of table 4.9. Our baseline "ProtoOnly" represents the result of further removing the seq2seq model, i.e., directly using the prototype as the output. We can see that

[4] We reported these fine-grained metrics only on Yelp dataset since its multiple (5) human references is crucial for making the results to match human judgments.

using prototype alone cannot yield satisfying performance and back-translation is crucial for better performances.

	InsR	InsAcc	DelR	DelAcc
CrossAligned	4.69	12.04	61.53	79.14
StyleEmbedding	1.32	2.30	32.80	36.05
MultiDecoder	3.38	6.13	52.07	63.26
DeleteOnly	8.52	14.34	60.56	73.29
DeleteAndRetrieve	9.09	14.39	61.07	73.47
ProtoOnly	**100.00**	20.50	**100.00**	75.09
AttributeEditor w/o BT	31.39	**28.79**	49.11	55.49
AttributeEditor(Ours)	49.02	28.29	74.57	**79.96**

Table 4.10: Ratio and accuracy of insertion and deletion on Yelp dataset.

InsR and DelR in table 4.10 demonstrate that our model has an apparent systematic bias towards editing the input sentence according to the prototype. And the higher InsAcc and DelAcc gained by our model confirm that using prototype is helpful for more accurate editing. We also notice that the advantage of our model is more salient for insert than for delete. This is probably because finding the correct words to add is generally harder than finding the correct words to delete.

Qualitative We showed sampled system outputs from Yelp dataset in table 4.11. From the table, we can see that our AttributeEditor model is competent at changing the attribute and preserving the independent content at the same time. The model can select the appropriate keywords (green words in table 4.11) from the prototype and use it the generation. Even some of the keywords from prototype do not match the human reference (second and fifth example in table 4.11) in term of automatic metrics; we can still verify their correctness by inspection.

We also notice that our model perform robustly well when the prototype does not contain any useful information. For the third and sixth example in table 4.11, we see the blue keywords are correctly chosen from the whole vocabulary instead of copying from the prototype.

Compared with two strong baseline methods (DeleteOnly and DeleteAndRetrieve), AttributeEditor makes less amount of edits on the input sentence, which are also more accurate than the edits made by other strong baselines. AttributeEditor also relies more on the retrieved prototype for its edits. All these observation is in accordance with the results shown in table 4.10.

Task	Data	Sentence
P2N	source	i ca n't believe how inconsiderate this pharmacy is .
	prototype	competent , and considerate !
	DeleteOnly	this pharmacy this is by far the best .
	DeleteAndRetrieve	this pharmacy is one of the best in madison .
	AttributeEditor	i ca n't believe how considerate this pharmacy is .
	reference	i ca n't believe how considerate this pharmacy is .
P2N	source	the decor was seriously lacking .
	prototype	seriously , beautiful modern decor .
	DeleteOnly	the place was clean and the decor was romantic !
	DeleteAndRetrieve	it was very clean and the decor was awesome .
	AttributeEditor	the decor was seriously beautiful .
	reference	the decor was seriously stunning .
P2N	source	but their inventory was questionable !
	prototype	love this store and their inventory .
	DeleteOnly	but their inventory was top notch and the food was delicious !
	DeleteAndRetrieve	but their inventory was the best in the longest !
	AttributeEditor	but their inventory was great !
	reference	but their inventory was great .
N2P	source	sit at the bar and eat the small plates .
	prototype	do n't sit at the bar ... .
	DeleteOnly	sit at the bar and eat the small plates .
	DeleteAndRetrieve	we could n't even sit at the bar and eat the small plates .
	AttributeEditor	do n't sit at the bar and eat small plates .
	reference	do n't sit at the bar for the small plates .
N2P	source	everyone who works there is always so happy and friendly .
	prototype	everyone who works there is busy .
	DeleteOnly	i could n't even everyone who works there there is so rude excuse .
	DeleteAndRetrieve	everyone who works there for everyone !
	AttributeEditor	everyone who works there is always so busy and not friendly .
	reference	everyone who works there is always so angry and unfriendly .
N2P	source	even their steamed veggies are amazing .
	prototype	steamed veggies were steamed to goo .
	DeleteOnly	even their steamed veggies , no sauce .
	DeleteAndRetrieve	even their steamed veggies were cold .
	AttributeEditor	even their steamed veggies are awful .
	reference	even their steamed veggies are awful .

Table 4.11: This table shows the hypothesis generated from our AttributeEditor system with the corresponding source input, prototype in the target sentiment, together with the outputs from strong baseline methods and human reference. "P2N" stands for positive to negative and "N2P" stands for negative to positive. Within hypothesizes, green words are the ones that appear in the prototype but not in the source input. Blue words are the those that do not appear either in the source or the prototype. In references, orange words are the words that appear in the reference but not in the source input.

Summary

We presented a novel approach for text attribute transfer in an unsupervised manner. By jointly modeling the input and prototype sentence, our model combines the power of search engine and seq2seq model. The combination convincingly led to the advantageous performance over representative baselines on three benchmark datasets.

Chapter 5

Conclusion

This thesis addresses several important challenges in learning cross-language and cross-style mappings when supervision is extremely limited. Specifically, by developing a set of unique approaches we overcome the limitations of existing works on model/word/sentence-level mapping learning as follows,

- **Semi-supervised learning of cross-language transfer of classification models**: We proposed to improve the transfer of classification models from high-resource languages to low-resource languages with two types of cross-lingual supervisions. Our first method uses a novel graph propagation algorithm to extend small bilingual dictionaries. The second method solves the ambiguity issue of cross-lingual transfer learning by generalizing a distillation algorithm on the parallel corpus. Both methods improved the state-of-art classification performance on cross-lingual text classification benchmarks.
- **Unsupervised learning of cross-language transfer of word embeddings**: Cross-lingual word embedding is crucial for many cross-language NLP tasks, but for many language pairs in various domains, large-quantity human-annotated training data are difficult to obtain. We developed a novel unsupervised approach to the joint learning of the bi-directional mappings between the word embedding distributions in the source language and the target language. This method achieves significantly better and more robust performance on extensive experiments of over 90 language pairs.
- **Semi-supervised and unsupervised learning for sentence-level style transfer**: Style and attribute transfer of text often suffers from the lack of human annotations for training. We developed a semi-supervised method which only requires a small amount of parallel data and achieved state-of-art performance on a formality transfer benchmark. We further address unsupervised attribute transfer with a novel pipeline that combines the strengths of a search engine and a Seq2Seq neural model. Empirical evaluations have demonstrated clear advantages of our methods over other state-of-art methods on formality and sentiment transfer benchmarks.

The overall contributions of this thesis advance the state-of-the-art for learning cross-language and cross-style mappings in multiple ways: First, we introduce a number of novel models for three representative mapping problems under limited or zero supervision. With more sophisticated objectives defined on unlabeled data, we address the limitations of previous works

and achieve the state-of-the-art on the benchmarks of each problem. Second, we demonstrate the generalizability of these techniques on multiple languages and styles, providing a comprehensive overview of the effectiveness and robustness of our methods. Third, our proposed models could be extended to other NLP tasks. For instance, CLDFA could be generalized to other sentence-level classification tasks such as textual entailment. And the unsupervised cross-lingual word embedding model we introduced could further enable cross-lingual transfer learning of a boarder range of NLP neural models. The semi-supervised and unsupervised SAT methods could be extended to other monolingual re-writing tasks, such as grammatical error correction or text simplification.

Promising directions for future works include:

- A general framework for the cross-lingual transfer of NLP models, where the choice of the source language, level of transfer, amount of supervision, and type of cross-lingual mapping could be automatically or systematically determined. For instance, a recent work [63] has shown promising results in this direction for choosing the source language automatically.
- Generalizing the transfer of non-contextualized word representations to the recent contextualized ones [19, 46, 81, 86, 121]. The pretrained contextualized word representation has improved upon the non-contextualized word representations and achieved the state-of-the-art for a wide range of monolingual NLP tasks. Since our proposed unsupervised cross-lingual word embedding model only requires samples of word representation from each of the languages, it could be extended to learn the mapping between two sets of contextualized word representations.
- Improving sentence-level mapping with large-scale pretraining. The pretraining on large unlabeled corpus has shown promising results on language generation. [20, 87, 121]. A natural extension would be to initialize our SAT model with the parameters pretrained on the large corpus and then fine-tune on the classification data and parallel data (if exists).
- Generalizing the number of domains from two to multiple. For instance, extending our cross-lingual model to multi-lingual setting, or our style/attribute transfer model to manipulate multiple aspects of the sentence in a single model.

Bibliography

[1] Hanan Aldarmaki, Mahesh Mohan, and Mona Diab. Unsupervised word mapping using structural similarities in monolingual embeddings. *Transactions of the Association for Computational Linguistics*, 6:185–196, March 2018. URL https://transacl.org/ojs/index.php/tacl/article/view/1317/292. 3.3.1

[2] Massih Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 28–36. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3690-learning-from-multiple-partially-observed-views-an-applicatio pdf. 2.1

[3] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*, 2016. 3.2

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. 70:214–223, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/arjovsky17a.html. 3.3.1

[5] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1250. URL https://www.aclweb.org/anthology/D16-1250. 3.1, 3.2, 3.3.2, 3.3, 3.3.2, 3.3.2

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1042. URL https://www.aclweb.org/anthology/P17-1042. 2.2.1, 2.2.1, 3.3, 3.3.2, 3.3, 3.3.2, 3.3.2

[7] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017. URL http://arxiv.org/abs/1710.11041. 3.1

[8] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1073. URL https://www.aclweb.org/anthology/P18-1073. 3.3

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473. 4.2, 4.2.2

[10] Nuria Bel, Cornelis H. A. Koster, and Marta Villegas. Cross-lingual text categorization. pages 126–139, 2003. 2.1

[11] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 3.3.2

[12] Hailong Cao, Tiejun Zhao, Shu Zhang, and Yao Meng. A distribution-based model to learn bilingual word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1818–1827, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1171. 3.3

[13] A P Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 1853–1861, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969034. 2.2.2, 2.2.3

[14] Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1627–1634, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1. URL http://dl.acm.org/citation.cfm?id=3042573.3042781. 2.3.4

[15] Xilun Chen, Ben Athiwaratkun, Yu Sun, Kilian Weinberger, and Claire Cardie. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*, 2016. 2.3.4, 2.3.4

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078186. 2.3.1

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2292–2300, USA, 2013. Curran Associates Inc. URL http:

//dl.acm.org/citation.cfm?id=2999792.2999868. 3.3, 3.3.1, 3.3.1

[18] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3079–3087. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.pdf. 2.3.1

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, June 2019. URL https://www.aclweb.org/anthology/N19-1423. 5

[20] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*, 2019. 5

[21] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 189–194, 2018. 4.2

[22] Kevin Duh, Akinori Fujino, and Masaaki Nagata. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 429–433, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-2075. 2.3

[23] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. pages 1285–1295, November 2016. doi: 10.18653/v1/D16-1136. URL https://www.aclweb.org/anthology/D16-1136. 3.2

[24] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1049. URL https://www.aclweb.org/anthology/E14-1049. 2.2.1, 3.2

[25] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 4.2, 4.2, 4.3.1, 4.3.1, 4.3.2

[26] Pascale Fung. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In *Third Workshop on Very Large Corpora*, 1995. URL https://www.aclweb.org/anthology/W95-0114. 3.3

[27] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045244. 2.3.3

[28] Tao Ge, Furu Wei, and Ming Zhou. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*, 2018. 4.2.1

[29] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1243–1252. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305510. 4.2.1

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf. 3.3, 3.3.1

[31] Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1157. URL https://www.aclweb.org/anthology/N15-1157. 3.2

[32] Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 748–756. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045199. 3.2

[33] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640, 2016. 4.2.2

[34] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 4.3.1

[35] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*, 2015. 4.2

[36] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5769–5779, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4. URL http://dl.acm.org/citation.cfm?id=3295222.3295327. 3.3.1

[37] Y. Guo and M. Xiao. Transductive representation learning for cross-lingual text classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 888–893, Dec 2012. doi: 10.1109/ICDM.2012.29. 2.1

[38] Yuhong Guo and Min Xiao. Cross language text classification via subspace co-regularized multi-view learning. pages 915–922, 2012. URL http://dl.acm.org/citation.cfm?id=3042573.3042692. 2.1

[39] Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018. doi: 10.1162/tacl_a_00030. URL https://www.aclweb.org/anthology/Q18-1030. 4.3.1, 4.3.1

[40] Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. pages 771–779, June 2008. URL https://www.aclweb.org/anthology/P08-1088. 3.3

[41] Tatsunori B. Hashimoto, Kelvin Guu, Yonatan Oren, and Percy Liang. A retrieve-and-edit framework for predicting structured outputs. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 10073–10083, USA, 2018. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=3327546.3327670. 4.3.1

[42] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 820–828, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL http://dl.acm.org/citation.cfm?id=3157096.3157188. 4.2

[43] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributed semantics. pages 58–68, June 2014. doi: 10.3115/v1/P14-1006. URL https://www.aclweb.org/anthology/P14-1006. 3.2

[44] Francis Heylighen and Jean-Marc Dewaele. Formality of language: definition, measurement and behavioral determinants. *Interner Bericht, Center "Leo Apostel", Vrije Universiteit Brüssel*, 1999. 4.2

[45] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531. 2.3

[46] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://www.aclweb.org/anthology/P18-1031. 5

[47] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596, 2017. 4.2, 4.2.1, 4.3.1

[48] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*,

pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1162. URL https://www.aclweb.org/anthology/P15-1162. 2.3.1

[49] Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. Unsupervised text style transfer via iterative matching and translation. *arXiv preprint arXiv:1901.11333*, 2019. 4.3.2

[50] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, 2015. 2.3.1

[51] Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 526–534. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045447. 2.3.1

[52] Yoon Kim. Convolutional neural networks for sentence classification. pages 1746–1751, October 2014. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181. 2.3.1, 4.2.1

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4.3.2

[54] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. pages 1459–1474, December 2012. URL https://www.aclweb.org/anthology/C12-1089. 2.2.2, 2.2.3

[55] Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. Learning bilingual word representations by marginalizing alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–229, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2037. URL https://www.aclweb.org/anthology/P14-2037. 3.2

[56] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2267–2273. AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2886521.2886636. 2.3.1

[57] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=rkYTTf-AZ. 4.3.1

[58] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. 2018. URL https://openreview.net/forum?id=H196sainb. (document), 3.1, 3.3, 3.3.2, 3.3.2, 3.3.2, 3.3, 9, 3.3.2, 3.4, 10

[59] Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ. 4.3.1

[60] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1005332.1005345. (document), 2.2.3, 2.2.3

[61] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, 2018. 4.2, 4.2.2, 4.3.1, 4.3.2, 4.3.2, 4.3.2

[62] Yiou Lin, Hang Lei, Jia Wu, and Xiaoyu Li. An empirical study on sentiment classification of Chinese review using word embedding. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 258–266, Shanghai, China, October 2015. URL https://www.aclweb.org/anthology/Y15-2030. (document), 2.3.4

[63] Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. Choosing transfer languages for cross-lingual learning. 2019. 5

[64] Xiao Ling, Gui-Rong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu. Can chinese web pages be classified with english data source? In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 969–978, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367628. URL http://doi.acm.org/10.1145/1367497.1367628. 2.1

[65] Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In Gregory Grefenstette, editor, *Cross-Language Information Retrieval*, pages 51–62. Springer US, Boston, MA, 1998. ISBN 978-1-4615-5661-9. doi: 10.1007/978-1-4615-5661-9_5. URL https://doi.org/10.1007/978-1-4615-5661-9_5. 2.3, 2.3.4

[66] Hanxiao Liu and Yiming Yang. Bipartite edge prediction via transductive learning over product graphs. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1880–1888, 2015. 2.2.1

[67] Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–256, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1028. URL https://www.aclweb.org/anthology/N15-1028. 3.1

[68] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752,

pages 41–48. Citeseer, 1998. 2.2.2

[69] Igor Melnyk, Cicero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. Improved neural text attribute transfer with non-parallel data. *arXiv preprint arXiv:1711.09395*, 2017. 4.2, 4.2, 4.3.1

[70] Antonio Valerio Miceli Barone. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. pages 121–126, August 2016. doi: 10.18653/v1/W16-1614. URL https://www.aclweb.org/anthology/W16-1614. 3.3

[71] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781. 2.2.3, 2.3.4, 3.1, 3.2, 3.3.2, 3.3, 3.3.2, 3.3.2

[72] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013. 2.2.1, 2.2.1

[73] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999792.2999959. 2.2.3

[74] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1090. 2.2.1

[75] Muhammad Tasnim Mohiuddin and Shafiq Joty. Revisiting adversarial autoencoder for unsupervised word translation with cycle consistency and improved training. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3857–3867, 2019. 3.3

[76] Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: Continuous revision of combinatorial structures. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2536–2544, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/mueller17a.html. 4.2, 4.3.1

[77] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2097. URL

https://www.aclweb.org/anthology/P15-2097. 4.2.2, 4.3.2

[78] Xing Niu, Sudha Rao, and Marine Carpuat. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1086. 4.2.2

[79] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135. 4.2.2, 4.3.2

[80] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162. 3.1

[81] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202. 5

[82] John C. Platt, Kristina Toutanova, and Wen-tau Yih. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 251–261, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1870658.1870683. 2.3, 2.3.4

[83] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:866–876, 2018. 4.3.1, 4.3.2

[84] Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1118–1127, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858681.1858795. (document), 2.3.4

[85] Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. Personalized machine translation: Preserving original author traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1074–1084, 2017. 4.2

[86] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving lan-

guage understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018. 5

[87] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019. 5

[88] Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 129–140, 2018. (document), 4.1, 4.2, 4.2.2, 4.2.2, 4.2.2, 4.2.2

[89] Reinhard Rapp. Identifying word translations in non-parallel texts. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 320–322, Cambridge, Massachusetts, USA, June 1995. Association for Computational Linguistics. doi: 10.3115/981658.981709. URL https://www.aclweb.org/anthology/P95-1050. 3.3

[90] Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing time-dependent data using dynamic t-sne. pages 73–77, 2016. doi: 10.2312/eurovisshort.20161164. URL https://doi.org/10.2312/eurovisshort.20161164. 2.3.3

[91] L. Rigutini, M. Maggini, and Bing Liu. An em based training algorithm for cross-language text categorization. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 529–535, Sep. 2005. doi: 10.1109/WI.2005.29. 2.1

[92] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019. 4.3.1

[93] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009. 4.2

[94] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96, 2016. 4.2.1, 4.2.2

[95] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725, 2016. 4.3.2

[96] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-*

guage Technologies, pages 35–40, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1005. URL https://www.aclweb.org/anthology/N16-1005. 4.2

[97] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf. 4.2, 4.2, 4.2.1, 4.3.1, 4.3.2, 4.3.2

[98] Lei Shi, Rada Mihalcea, and Mingjun Tian. Cross language text classification by model translation and semi-supervised learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1057–1067, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D10-1103. 2.1, 2.2.2, 2.2.3

[99] Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Ridge regression, hubness, and zero-shot learning. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 135–151, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23528-8. 3.2, 3.3.2, 3.3, 3.3.2, 3.3.2

[100] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. 2017. URL https://openreview.net/forum?id=r1Aab85gg. 2.2.1, 2.2.1, 3.1

[101] Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. Inverted indexing for cross-lingual NLP. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1713–1722, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1165. URL https://www.aclweb.org/anthology/P15-1165. 3.1, 3.2

[102] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf. 2.3.1

[103] Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. UM-corpus: A large English-Chinese parallel corpus for statistical machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1837–1842, Reykjavik, Iceland, May 2014. European Languages Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/

pdf/774_Paper.pdf. 2.3.4

[104] Youzhi Tian, Zhiting Hu, and Zhou Yu. Structured content preservation for unsupervised text style transfer. *arXiv preprint arXiv:1810.06526*, 2018. 4.2, 4.3.1

[105] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012. 3.3.1

[106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 4.2.1, 4.2.2, 4.3.1, 4.3.1, 4.3.2

[107] G Vinodhini and RM Chandrasekaran. Sentiment analysis and opinion mining: A survey. *International Journal*, 2(6):282–292, 2012. 4.1

[108] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 1497–1504, Cambridge, MA, USA, 2002. MIT Press. URL http://dl.acm.org/citation.cfm?id=2968618.2968804. 2.3, 2.3.4

[109] Ivan Vulić and Anna Korhonen. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 247–257, Berlin, Germany, 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1024. 3.2

[110] Ivan Vulic and Marie-Francine Moens. Bilingual distributed word representations from document-aligned comparable data. *J. Artif. Int. Res.*, 55(1):953–994, January 2016. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=3013558.3013583. 3.2

[111] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 235–243, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL http://dl.acm.org/citation.cfm?id=1687878.1687913. 2.1

[112] Min Xiao and Yuhong Guo. A novel two-step method for cross language representation learning. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1259–1267. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5164-a-novel-two-step-method-for-cross-language-representation-lea
pdf. 2.3, 2.3.4, 2.3.4

[113] Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379,

Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1034. 3.1

[114] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1104. URL https://www.aclweb.org/anthology/N15-1104. 3.2, 3.3.2, 3.3, 3.3.2, 3.3.2

[115] Jingjing Xu, SUN Xu, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, 2018. 4.2, 4.2, 4.3.1, 4.3.2

[116] Ruochen Xu and Yiming Yang. Cross-lingual distillation for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, 2017. 1

[117] Ruochen Xu, Yiming Yang, Hanxiao Liu, and Andrew Hsi. Cross-lingual text classification via model translation with limited dictionaries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 95–104. ACM, 2016. 1

[118] Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474, 2018. 1, 1

[119] Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. Paraphrasing for style. pages 2899–2914, December 2012. URL https://www.aclweb.org/anthology/C12-1177. 4.2

[120] Pengcheng Yang, Fuli Luo, Shuangzhi Wu, Jingjing Xu, Dongdong Zhang, and Xu Sun. Learning unsupervised word mapping by maximizing mean discrepancy. *arXiv preprint arXiv:1811.00275*, 2018. 3.3

[121] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 5

[122] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. pages 1480–1489, June 2016. doi: 10.18653/v1/N16-1174. URL https://www.aclweb.org/anthology/N16-1174. 2.3.1

[123] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7287–7298. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7959-unsupervised-text-style-transfer-using-language-models-as-dis

pdf. 4.2, 4.3.1

[124] Pengcheng Yin, Graham Neubig, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Learning to represent edits. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJl6AjC5F7. 4.3.1

[125] Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1160. URL https://www.aclweb.org/anthology/D16-1160. 4.2

[126] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1179. URL https://www.aclweb.org/anthology/P17-1179. (document), 3.1, 3.3, 3.3.2, 3.3.2, 3.3, 9, 3.3.2, 3.3.2

[127] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1207. URL https://www.aclweb.org/anthology/D17-1207. 3.3, 3.3.1

[128] Rui Zhang, Honglak Lee, and Dragomir R. Radev. Dependency sensitive convolutional neural networks for modeling sentences and documents. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1512–1521, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1177. URL https://www.aclweb.org/anthology/N16-1177. 2.3.1

[129] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969312. (document), 2.3.4

[130] Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1108, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1138. 4.2, 4.3.1

[131] Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. Ten pairs to tag – multilingual POS tagging via coarse mapping between embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, pages 1307–1317, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1156. URL https://www.aclweb.org/anthology/N16-1156. 3.2, 3.3.2, 3.3, 3.3.2, 3.3.2

[132] Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018. 4.3.1

[133] Chunting Zhou, Xuezhe Ma, Di Wang, and Graham Neubig. Density matching for bilingual word embedding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1588–1598, 2019. 3.3