Addressee and Response Selection for Multilingual Conversation

Motoki Sato[1*] **Hiroki Ouchi**[2,3] **Yuta Tsuboi**[1*]
[1]Preferred Networks, Inc.,
[2]RIKEN Center for Advanced Intelligence Project
[3]Tohoku University
sato@preferred.jp, hiroki.ouchi@riken.jp, tsuboi@preferred.jp

Abstract

Developing conversational systems that can converse in many languages is an interesting challenge for natural language processing. In this paper, we introduce *multilingual addressee and response selection*. In this task, a conversational system predicts an appropriate addressee and response for an input message in multiple languages. A key to developing such multilingual responding systems is how to utilize high-resource language data to compensate for low-resource language data. We present several knowledge transfer methods for conversational systems. To evaluate our methods, we create a new multilingual conversation dataset. Experiments on the dataset demonstrate the effectiveness of our methods.

1 Introduction

Open-domain conversational systems, such as chatbots, are attracting a vast amount of interest and play their functional and entertainment roles in real-world applications. Recent conversational models are often built in an end-to-end fashion using neural networks, which require a large amount of training data (Vinyals and Le, 2015; Serban et al., 2016). However, it is challenging to collect enough data to build such models for many languages. Consequently, most work has targeted high-resource languages, such as English and Chinese (Shang et al., 2015; Serban et al., 2016).

In this work, we aim to develop *multilingual* conversational systems that can return appropriate responses in many languages. Specifically, we assume the two types of systems: (i) **language-specific** systems and (ii) **language-invariant** systems. A language-specific system consists of multiple conversational models, each of which returns responses in a corresponding language. By contrast, a language-invariant system consists of a single unified model, which returns responses in all target languages. A key to building these multilingual models is how to utilize high-resource language data to compensate for low-resource language data. We present several knowledge-transfer methods. To the best of our knowledge, this is the first work focusing on low-resource language enablement of conversational systems.

One challenge when developing conversational systems is how to evaluate the system performance. For generation-based conversational systems, which generate each word for a response one by one, many studies adopt human judgments. However, it is costly and impractical to adopt this evaluation method for multilingual systems, especially for minor-language systems. Thus, as a first step, we develop retrieval-based conversational systems and evaluate the ability to select appropriate responses from a set of candidates. Fig. 1 shows the overview of our multilingual responding systems.

This paper provides: (i) formal task definitions, (ii) several knowledge-transfer methods and (iii) a multilingual conversation dataset. First, we introduce and formalize the two task settings: *single-language adaptation* for language-specific systems and *multi-language adaptation* for language-invariant systems (Sec. 4). Second, we present several methods leveraging high-resource language data to compensate for low-resource language in the two settings (Sec. 5). Our basic method uses multilingual word embeddings and transfers source-language knowledge to target languages (Sec. 5.1 (a)). We also design

*This work was conducted when the first author worked at Nara Institute of Science and Technology, and portions of this research were done while the third author was at IBM Research - Tokyo.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3631–3644
Santa Fe, New Mexico, USA, August 20-26, 2018.

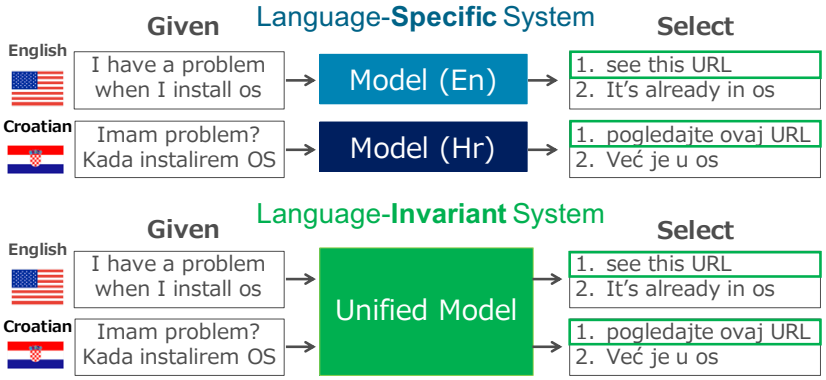

Figure 1: Example of multilingual response selection.

three extended methods. Among them, the fine-tuning method builds a model specific to a single target language (Sec. 5.2 (b)). The joint loss training and the multi-language adversarial training methods build a unified model invariant for multiple target languages (Sec. 5.2 (c) and (d)). Third, we create a multilingual conversation corpus and dataset[1] (Sec. 6). From the Ubuntu IRC Logs[2], we collect the logs in 12 languages.

To show benchmark results, we perform experiments on the created dataset (Sec. 7). The results demonstrate that our methods allow models to effectively adapt to low-resource target languages. In particular, our method using Wasserstein GAN (Arjovsky et al., 2017) achieves high-performance for simultaneously dealing with multiple languages with a single unified model.

2 Related Work

Short Text Conversation

In short text conversation, a system predicts an appropriate response for an input message in *single-turn, two-party conversation* (Ritter et al., 2011). One major approach to it is the generation-based approach, which generates a response using a sequence-to-sequence model (Shang et al., 2015; Vinyals and Le, 2015; Serban et al., 2016; Li et al., 2016; Mei et al., 2017). Another popular approach is the retrieval-based approach, which retrieves candidate responses from a repository and returns the highest scoring one using a ranking model (Wang et al., 2013; Lu and Li, 2013; Ji et al., 2014; Wang et al., 2015). Lowe et al. (2015) proposed next utterance classification (NUC), in which a model has to select an appropriate response from a fixed set of candidates.

Evaluation for Conversational Systems

Evaluation methods for conversational systems are an open question (Lowe et al., 2015; Liu et al., 2016; Lowe et al., 2017). While many of previous studies on conversational systems used human judgements, automatic evaluation methods are attractive because it is much easier and less costly to use. However, according to Liu et al. (2016), for generation-based systems, automatic evaluation metrics, such as BLEU (Papineni et al., 2002), correlate very weakly with human judgements.

For retrieval-based systems, some studies used ranking-based metrics, such as mean average precision and accuracy (Ji et al., 2014; Wang et al., 2015). Lowe et al. (2016) confirmed the feasibility of NUC as a surrogate task for building conversational systems. Although there are controversial issues for these evaluation methods (Lowe et al., 2015), as a practical choice, we adopt the accuracy-based metric for evaluating multilingual conversational systems.

[1]Our code and dataset are publicly available at `https://github.com/aonotas/multilingual_ASR`
[2]http://irclogs.ubuntu.com/

Addressee and Response Selection

NUC focuses on two-party, *multi-turn conversation*. As an extension of it, Ouchi and Tsuboi (2016) proposed addressee and response selection (ARS) for *multi-party conversation*. ARS integrates the addressee detection problem, which has been regarded as a problematic issue in multi-party conversation (Traum, 2003; Jovanović and Akker, 2004; Bohus and Horvitz, 2011; Uthus and Aha, 2013). Mainly, this problem has been tackled in spoken/multimodal dialog systems (Jovanović et al., 2006; Akker and Traum, 2009; Nakano et al., 2013; Ravuri and Stolcke, 2014). While these systems largely rely on acoustic signal or gaze information, ARS focuses on text-based conversations. Extending these studies, we tackle multilingual, multi-turn, and multi-party text conversation settings.

Cross-Lingual Conversation

The motivation of our task is similar with that of Kim et al. (2016). They tackled cross-lingual dialog state tracking in English and Chinese. While they transfer knowledge from English to Chinese, we transfer knowledge between a high-resource and several low-resource languages.

3 Addressee and Response Selection

Addressee and response selection (ARS)[3], proposed by Ouchi and Tsuboi (2016), assumes the situation where a responding agent returns a response to an addressee following a conversational context.

Formally, given an input conversational situation $\boldsymbol{x} \in X$, a system predicts $\boldsymbol{y} \in Y$, which consists of an addressee a and a response $\boldsymbol{r}$:

$$\textsc{Given} : \boldsymbol{x} = (a_{\mathrm{res}},\ \mathcal{C},\ \mathcal{R}), \quad \textsc{Predict} : \boldsymbol{y} = (a, \boldsymbol{r})$$

where a_{res} is a responding agent, $\mathcal{C}$ is a context (a sequence of previous utterances) and $\mathcal{R}$ is a set of candidate responses. To predict an addressee a, we select an agent from a set of the agents appearing in a context $\mathcal{A}(\mathcal{C})$. To predict a response $\boldsymbol{r}$, we select a response from a set of candidate responses $\mathcal{R} = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{|\mathcal{R}|}\}$.

This task evaluates accuracy on the three aspects: addressee-response pair selection (`ADR-RES`), addressee selection (`ADR`), and response selection (`RES`). In `ADR-RES`, we regard the answer as correct if both the addressee and response are correctly selected. In `ADR`/`RES`, we regard the answer as correct if the addressee/response is correctly selected.

4 Multilingual Addressee and Response Selection

As an extension of monolingual ARS, we propose *multilingual addressee and response selection* (M-ARS). In ARS, a system is given as input a set of candidate responses and a conversational context in a single language. By contrast, in M-ARS, a system receives the inputs in one of multiple languages. In the following, we first explain our motivation for tackling M-ARS and then describe the formal task definitions.

4.1 Motivative Situations

We assume the two multilingual conversational situations:

- You want to build *language-specific* systems, each of which responds in a single language.
- You want to build one *language-invariant* system, which responds in multiple languages.

The first situation is that we build K models, each of which is specialized for one of K target languages. The second one is that we build one unified model that can deal with all K target languages. Taking these situations into account, we present the corresponding two tasks: (i) *single-language adaptation* and (ii) *multi-language adaptation*.

[3]Due to the space limitation, we give a brief overview of ARS. For the complete task definition, please refer to Ouchi and Tsuboi (2016).

4.2 Task Overview

The goal of *single-language adaptation* is to develop and evaluate a language-specific ARS model for a single target language. For example, using English, German and Italian training data, we build a model specialized for German conversation. The goal of *multi-language adaptation* is to develop and evaluate a language-invariant ARS model for multiple target languages. For example, using English, German and Italian training data, we build a model that can respond to not only German but also Italian and English conversation. In the following subsections, we formalize each of these tasks.

4.3 Formal Task Definition

We assume that we have conversation data in each of a set of languages $\mathcal{K}$.

Training
In the training phase, a training dataset is given for each language $k \in \mathcal{K}$:

$$\mathcal{D}_{\text{train}}^{(k)} = \{\, \boldsymbol{x}_i^{(k)}, \boldsymbol{y}_i^{(k)} \,\}_{i=1}^{N^{(k)}}, \;\; k \in \mathcal{K}$$
$$\mathcal{D}_{\text{train}} = \bigcup_k \mathcal{D}_{\text{train}}^{(k)}$$

where $\boldsymbol{x}^{(k)}$ and $\boldsymbol{y}^{(k)}$ are a conversational situation and the target output in language k, respectively. We train a model $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ on these training samples.

Evaluation
In single-language adaptation, we evaluate a trained model for a single target language $t \in \mathcal{K}$. The trained model receives an input of the target language, $\boldsymbol{x}^{(t)} \sim \mathcal{D}_{\text{eval}}^{(t)}$, and predicts $\hat{\boldsymbol{y}}^{(t)}$. As evaluation metrics, we use the three accuracies (`ADR-RES`, `ADR` and `RES`) used in ARS (Sec. 3).

In multi-language adaptation, given evaluation datasets for all the languages $\mathcal{K}$, i.e., $\bigcup_k \mathcal{D}_{\text{eval}}^{(k)}$, the trained model receives an input of each language $\boldsymbol{x}^{(k)} \sim \mathcal{D}_{\text{eval}}^{(k)}$ and predicts $\hat{\boldsymbol{y}}^{(k)}$. As evaluation metrics, we use macro average over all the languages: $\texttt{ADR-RES} = \frac{\sum_k \texttt{ADR-RES}^{(k)}}{|\mathcal{K}|}$. `ADR` and `RES` are also computed in the same way.

5 Methods

In this section, we firstly describe a model used for addressee and response selection, and then explain our proposed methods to train parameters of the model.

Our model $\mathcal{F}$ consists of a feature extractor f^E, addressee scoring function f^A and response scoring function f^R. f^A and f^R return relevance scores (probabilities) for an addressee and response:

$$f^A(\boldsymbol{x}, a_i) = \sigma([\mathbf{a}_{\text{res}}, \mathbf{h}_c]^{\mathrm{T}}\, \mathbf{W}_a\, \mathbf{a}_i) \quad (1)$$
$$f^R(\boldsymbol{x}, \boldsymbol{r}_j) = \sigma([\mathbf{a}_{\text{res}}, \mathbf{h}_c]^{\mathrm{T}}\, \mathbf{W}_r\, \mathbf{r}_j) \quad (2)$$

where $\mathbf{a}_{\text{res}}$ is a responding agent vector, $\mathbf{h}_c$ is a conversational context vector, $\mathbf{a}_i$ is an agent vector, and $\mathbf{r}_j$ is a candidate response vector. All these vectors are encoded by the feature extractor f^E. We use the dynamic model (Ouchi and Tsuboi, 2016) as f^E. Fig. 2 shows the overview of the dynamic model. This model represents each agent as a hidden state vector that dynamically changes along with time steps in GRU (Cho et al., 2014). [4]

A model $\mathcal{F}$ is parameterized by $\theta = \{\theta_E \cup \{\mathbf{W}_a, \mathbf{W}_r\}\}$, where θ_E is parameters of f^E. To train these parameters, we present four methods. These methods assume that we have training sets for a set of languages $\mathcal{K}$: some of them are high-resource languages $\mathcal{S} \subseteq \mathcal{K}$ and others are relatively low-resource languages $\mathcal{T} = \bar{\mathcal{S}}$.

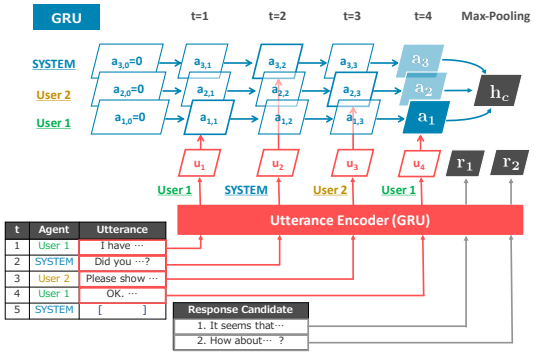

Figure 2: Overview of Dynamic Model.

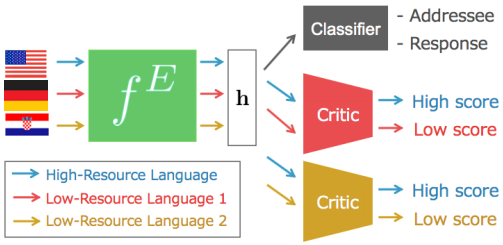

Figure 3: Overview of our W-GAN training method for multiple target languages.

5.1 A Basic Method

(a) Multilingual Embedding Replacement

This method trains a model $\mathcal{F}$ on high-resource language data $\mathcal{D}^{(s)}_{\text{train}}$, where $s \in \mathcal{S}$, and uses the trained model for responding conversations in other languages $\bar{\mathcal{S}}$. To realize this transfer, we use *multilingual embeddings*.

Consider the case where the high-resource language is English (En) and low-resource language is German (De). In the training phase, we use English word embeddings to train model parameters.[5] In the testing phase, instead of the English embeddings, we use German embeddings:

$$\textbf{Train: } \mathbf{w} = \mathbf{W}^{(\texttt{En})}_{\text{emb}}\, w \qquad \textbf{Test: } \mathbf{w} = \mathbf{W}^{(\texttt{De})}_{\text{emb}}\, w$$

where w is a one-hot vector. We just replace the English word embeddings $\mathbf{W}^{(\texttt{En})}_{\text{emb}}$ with the German ones $\mathbf{W}^{(\texttt{De})}_{\text{emb}}$. After looking up each word embedding $\mathbf{w}$, the neural model computes the hidden states. One advantage of this method is to require no target language data. As multilingual embeddings, we use MultiCCA[6] proposed by Ammar et al. (2016). In these embeddings, semantically similar words in the same language or translationally equivalent words in different languages are projected onto nearby.

Besides multilingual embeddings, another option to build a conversational model without no conversation data in a target language is to translate high-resource language data to low-resource one and train a conversational model on the translations. One limitation of this approach is that it is costly to prepare parallel corpora for building the translation model. We discuss this approach in Sec. 7.3.

5.2 Extended Methods

We present the two types of methods which use target language data for building (i) *language-specific models* and (ii) *language-invariant models*.

5.2.1 Methods for Language-Specific Models

(b) Fine-Tuning with Target Language Data

To compensate for the lack of the low-resource language data, this method firstly trains a model $\mathcal{F}_\theta$ on high-resource language data (pre-training phase). Then, using the pre-trained parameters θ as the initial values, this method re-trains them on low-resource language data (fine-tuning phase). We can expect that by gaining the better initial parameters, the tuning effectively adapts the model to the target language.

[4] In this example, a responding agent vector $\mathbf{a}_{\text{res}}$ is $\mathbf{a}_3$. Note that the states of the agents that are not speaking at the time are updated by zero vectors.

[5] The embeddings are fixed, not fine-tuned, during training.

[6] The pre-trained MultiCCA embeddings are provided at http://128.2.220.95/multilingual/data/

5.2.2 Methods for Language-Invariant Models

In order to build language-invariant models, it is critical to consider the two perspectives: (i) *avoiding catastrophic forgetting* and (ii) *learning language-invariant features*. Catastrophic forgetting (Kirkpatrick et al., 2017) is the phenomenon that a model forgets knowledge of previously trained tasks (languages) by incorporating knowledge of the current task (language). Language-invariant features are the features that are common and unchanged in different languages. Taking these two perspectives into account, we present the following two methods.

(c) Joint Loss Training

This method aims to avoid catastrophic forgetting by jointly training model parameters on all the language data at a time. Assuming that we have a set of languages $\mathcal{K}$, this method uses the joint loss function: $\mathcal{J}_{\text{joint}}(\theta) = \sum_k \mathcal{J}(\mathcal{D}^{(k)}, \theta)$ where the loss function $\mathcal{J}$ is the cross-entropy loss used in Ouchi and Tsuboi (2016).

(d) Multi-Language Adversarial Training

To learn language-invariant features, we use a framework of Wasserstein-GAN (W-GAN) (Arjovsky et al., 2017), a recently proposed technique to improve stability for generative adversarial nets (GANs) (Goodfellow et al., 2014). The aim of this method is to match the distributions of feature representations in two languages.

Fig. 3 illustrates an example. English is the high-resource language $s \in \mathcal{S}$, and German and Croatian are low-resource languages $t \in \mathcal{T}$. For each language, the feature extractor f^E receives an input conversation $\boldsymbol{x}$ and computes the hidden features $\mathbf{h} = f^E(\boldsymbol{x})$[7]. Thus, by using f^E, we obtain the hidden feature $\mathbf{h}^{(s)}$ and $\mathbf{h}^{(t)}$ for English and the others, respectively.

A pair of the high- and low-resource language features $\mathbf{h}^{(s)}$ and $\mathbf{h}^{(t)}$ is given to a critic g_π to minimize the Wasserstein distance between the distributions $p(\mathbf{h}^{(s)})$ and $p(\mathbf{h}^{(t)})$:

$$\mathcal{W}(\, p(\mathbf{h}^{(s)}),\, p(\mathbf{h}^{(t)})\,) = \max_{\pi}\, \mathbb{E}_{\mathbf{h}^{(s)}\sim p(\mathbf{h}^{(s)})}[g_\pi(\mathbf{h}^{(s)})] - \mathbb{E}_{\mathbf{h}^{(t)}\sim p(\mathbf{h}^{(t)})}[g_\pi(\mathbf{h}^{(t)})] \tag{3}$$

where the maximum is taken over the set of all 1-Lipschitz functions g_π.[8] By maximizing this equation, the distributions of the feature representations, $p(\mathbf{h}^{(s)})$ and $p(\mathbf{h}^{(t)})$, are made as close as possible. In this paper, as the critic g_π, we use multi-layer perceptron.

Eq. 3 is designed for the two distributions. Thus, we generalize this W-GAN equation to deal with $|\mathcal{S}|$ high-resource languages and $|\mathcal{T}|$ low-resource languages:

$$\mathcal{J}_{\text{wgan}}(\theta) = \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \mathcal{W}(\, p(\mathbf{h}^{(k)}),\, p(\mathbf{h}^{(\ell)})\,)$$

This loss function $\mathcal{J}_{\text{wgan}}$ is integrated with the joint loss: $\mathcal{J}_{\text{adv}}(\theta) = \mathcal{J}_{\text{joint}}(\theta) + \lambda\, \mathcal{J}_{\text{wgan}}(\theta)$ where λ is a hyper-parameter that balances these loss functions and we used $\lambda = 0.5$.

6 Corpus and Dataset

One of our goals is to provide a multilingual conversation corpus/dataset that can be used over a wide range of conversation research. We follow the corpus and data creation method of Ouchi and Tsuboi (2016). First, we crawl the Ubuntu IRC Logs[9], and preprocess the logs in many languages. Each language is identified by using a language detection library (Nakatani, 2010). The resulting corpus consists of multilingual conversations in 12 languages, shown in Tab. 1.

Then, we create an M-ARS dataset. For each language, we set the ground-truth/false addressees and responses following Ouchi and Tsuboi (2016). Note that the addressed usernames in utterances have been

[7] Hidden feature representation $\mathbf{h}$ is the concatenation of the responding speaker vector and context vector in Eqs. 1 and 2, i.e. $\mathbf{h} = [\mathbf{a}_{\text{res}}, \mathbf{h}_c]$.

[8] A function g is 1-Lipschitz when $|g(x) - g(y)| \leq |x - y|$ for all x and y. To constrain the critic g to a 1-Lipschitz function, the parameters of g are clipped to a fixed range.

[9] We use a collection of the logs during one year (2015). We plan to expand it by collecting the logs over all the years.

Language	Corpus Docs	Utters	Words
English (en)	7355	2.4 M	27.0 M
Italian (it)	357	165 k	1.1 M
Croatian (hr)	254	80 k	630 k
German (de)	248	38 k	335 k
Portuguese (pt)	211	52 k	285 k
Slovenian (sl)	179	59 k	357 k
Polish(pl)	67	8.8 k	51 k
Dutch (nl)	57	7.2 k	75 k
Spanish (es)	36	7.1 k	49 k
Swedish (sv)	26	1.7 k	6.8 k
Russian (ru)	5	0.3 k	1.5 k
French (fr)	3	0.5 k	3.0 k

Table 1: Statistics of M-ARS corpus.

Language	Dataset Train	Dev	Test
English (en)	665.6 k	45.1 k	51.9 k
Italian (it)	38,511	2,561	3,873
Croatian (hr)	11,387	512	1,145
German (de)	5,500	354	569
Portuguese (pt)	5,951	285	975

Table 2: Statistics of the M-ARS dataset.

Setting	Method	$\|\mathcal{R}\| = 2$ ADR−RES	ADR	RES	$\|\mathcal{R}\| = 10$ ADR−RES	ADR	RES
Single Language Adaptation	CHANCE	3.97	7.94	50.00	0.80	7.94	10.00
	TF-IDF	39.51	64.97	60.61	12.54	64.97	18.50
	TRGONLY	47.35	69.27	67.35	19.42	69.73	26.13
	ENONLY	38.07	65.72	57.65	8.50	62.38	13.75
	FINETUNE	49.58	69.59	69.84	21.15	70.33	28.15
	JOINT	51.55	70.30	71.88	22.32	**70.36**	29.38
	WGAN	**53.17**	**70.99**	**73.25**	**23.34**	70.20	**30.39**
Two Language Adaptation	CHANCE	2.30	4.59	50.00	0.46	4.59	10.00
	TF-IDF	38.32	60.29	64.25	13.99	60.29	23.84
	ENONLY	46.77	67.62	67.90	19.88	65.86	27.83
	FINETUNE	50.98	68.79	72.60	24.30	68.89	32.96
	JOINT	53.37	69.75	74.94	26.60	69.75	35.59
	WGAN	**54.14**	**70.07**	**75.63**	**27.23**	**69.76**	**36.11**
Five Language Adaptation	TF-IDF	39.04	63.10	62.06	13.12	63.10	20.64
	ENONLY	41.55	66.48	61.75	13.05	63.77	19.38
	JOINT	50.69	69.00	72.18	22.80	69.18	31.11
	WGAN	**52.11**	**69.74**	**73.34**	**23.39**	**69.35**	**31.88**

Table 3: Results for Single-/Two-/Five- language adaptation. Each number represents accuracy on addressee-response selection (ADR−RES), addressee selection (ADR) or response selection (RES).

removed for addressee selection. Thus, we have to predict the addressees without seeing the addressed usernames. The number of candidate responses ($|\mathcal{R}|$) is set to 2 or 10. The dataset is then randomly partitioned into a training set (90%), a development set (5%) and a test set (5%). Tab. 2 shows the statistics of the top 5 largest language sections of this resulting dataset.

7 Experiments

7.1 Experimental Setup

7.1.1 Task Settings

We use the following languages: (i) English (En) as the high-resource language, and (ii) Italian (It), Croatian (Hr), German (De), Portuguese (Pt) as the low-resource languages. In the following, we describe the languages used in each task.

(a) Single-Language Adaptation

Train: English + 1 Low-Res. Language, **Dev & Test:** 1 Low-Res. Language

For example, in the case that the target is Italian (It), we use the En and It training sets to train a model, and evaluate the trained model on the It test set. As evaluation metrics, we use the three types

of accuracies, ADR−RES, ARD and RES (described in Sec. 3). We report the macro average accuracies of all source-target language pairs (En-It, En-Hr, En-De, and En-Pt).

(b) Multi-Language Adaptation

Train: English + $|\mathcal{T}|$ Low-Res. Languages, **Dev & Test:** English + $|\mathcal{T}|$ Low-Res. Languages

We use the En, It, Hr, Pt and De training sets to train a unified model, and evaluate it on the test sets for all the languages (En, It, Hr, Pt, De). As evaluation metrics, we use the macro averages of ADR−RES, ARD and RES for all the languages. For example, for two language adaptation ($|\mathcal{T}| = 1$), we report the macro averages over all the language pairs (En-It, En-Hr, En-De, and En-Pt). For five language adaptation ($|\mathcal{T}| = 4$), we report the macro averages over all the five languages. Note that while we evaluate the performance on only the test set of the target low-resource language in single-language adaptation, we evaluate it on the test sets of English and the low-resource languages in multi-language adaptation.

7.1.2 Comparative Methods

We compare several methods. Our proposed methods (Sec. 5) are orthogonal, so that we can combine a method with others. In the following, we list the methods used in the comparison.

- TRGONLY: A dynamic model proposed by Ouchi and Tsuboi (2016) trained on only the low-resource target language data.
- ENONLY: A model built by (a) *multilingual embedding replacement* in Sec. 5.1: training a model on the English data and replacing the English word embeddings with the embeddings of the low-resource language.
- FINETUNE: A model built by (b) *fine-tuning* in Sec. 5.2: training a model on the high-resource language (English), and retraining it on the low-resource language.
- JOINT: A model built by (b) *fine-tuning* and (c) *joint loss training*: building a model by FINETUNE as an initial model, and retraining it with the joint loss functions.
- WGAN: A model built by (b) *fine-tuning* and (d) *multi-language adversarial training*: building a model by FINETUNE as an initial model, and retraining it with W-GAN.

Besides the neural models, we also use the TF-IDF model used in Ouchi and Tsuboi (2016). This model firstly creates TF-IDF vectors for the context and each candidate response. Then, it computes the cosine similarity for each pair of the context vector and a response vector. Finally, it selects the candidate response with the highest similarity.

7.1.3 Optimization

We use stochastic gradient descent (SGD) with a mini-batch method. To update parameters, we use *Adam* (Kingma and Ba, 2014). We describe the details of hyper-parameter settings in Supplementary Material.

7.2 Results

Tab. 3 shows the results of single-language and multi-language adaptation. Note that Tab. 6, Tab. 7, and Tab. 8 shows the detailed results for each language.

Single-Language Adaptation

WGAN achieved the best scores for most of the metrics. This suggests that the W-GAN method successfully transfers knowledge of high-resource language to a low-resource language. Also, FINETUNE outperformed TRGONLY. This means that pre-training parameters on the high-resource language data improves a model for a target low-resource language. Interestingly, ENONLY achieved higher scores than chance-level without any target language data. One possible explanation is that the multilingual embeddings have good alignments to some extent between similar meaning words in different languages.

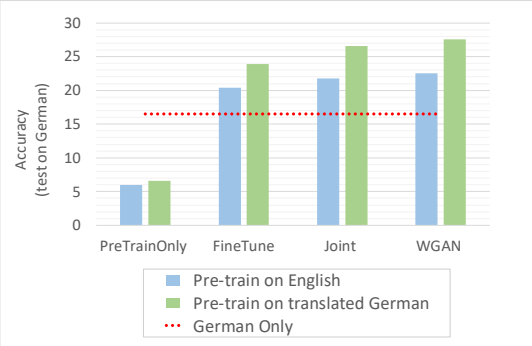

Figure 4: Effects of data augmentation with NMT.

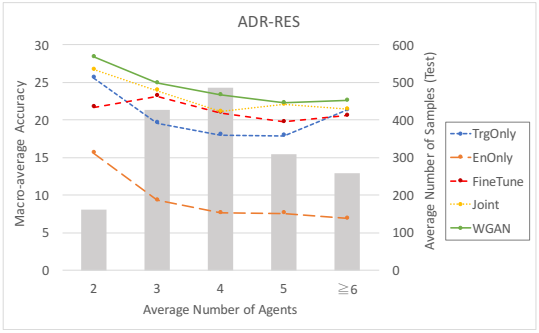

Figure 5: Effects of the number of agents in the context. Left axis: ADR-RES accuracy on test sets (drawn as lines). Right axis: the average number of test samples (drawn as bars).

Multi-Language Adaptation

In both two- and five-language adaptation, WGAN achieved the best scores. Specifically, In five language adaptation, regardless of using a single, unified model, JOINT and WGAN achieved high-performance.[10] Also, WGAN outperformed JOINT in all the metrics. This suggests that WGAN learns language-invariant features more effectively.

In NLP tasks, Chen et al. (2016) applied W-GAN to cross-lingual sentiment classification and successfully transferred the source-side knowledge to the target one. In this paper, we have extended the adaptation of single source-target pair to the adaptation of multiple pairs. Our experimental results show that our method works well for multi-language adaptation in conversation domain.

7.3 Data Augmentation with NMT

As we mentioned in Sec. 5.1 (a), as another approach to compensating for low-resource language, we use data augmentation. To increase the amount of the training set of a low-resource language, we translate high-resource language (English) samples to low-resource ones by using Neural Machine Translation (NMT). Although some translations are noisy, we can obtain much more training samples for a low-resource language. One limitation of this method is that it is costly to prepare parallel corpora, which is often unavailable for low-resource languages. For reproducibility, we use publicly available NMT models already trained on a parallel corpus. Since OpenNMT[11] provides an English-German model, we conduct experiments for the English-German pair.

We investigate the effects of translated German data for pre-training a model. Translating English (En) training utterances to German (De) ones by using the trained NMT model, we can obtain translated German training data (De′). We compare the two settings: (i) pre-training a model on English data and (ii) pre-training a model on translated German data. After pre-training, we apply the methods used in Sec. 7. We evaluate the performance with ADR-RES accuracy on the German test set.

Fig. 4 shows the results ($|\mathcal{R}| = 10$). The red dotted line is the performance of the modes trained on only original German training data (De). In each method, the blue bar at left hand is a model pre-trained on English (En), and the green bar at right hand is a model pre-trained on (De′).

PRETRAINONLY, the left-most method in Fig.4, uses the pre-trained models. The results were almost the same between models pre-trained on English or translated German data, and worse than the model trained on only the original German training data (red dotted line). This suggests that only the translations by NMT are not sufficient for building good multilingual ARS models.

Furthermore, we re-train the pre-trained models by using the three methods, FINETUNE, JOINT and WGAN. In other words, each method uses a pre-trained model as the initial model and re-trains the parameters. In all methods, the models re-trained from the De′ pre-trained model (green bars) were better

[10] Since FINETUNE builds a model for each target language, it cannot analyze multiple languages with a single model. That is why there is no result of FINETUNE in five-language adaptation.

[11] http://opennmt.net/Models/

than the ones from the En pre-trained ones (blue bars). This suggests that by combining the NMT-based data augmentation method with the knowledge-transfer methods, the performance is boosted. Another point is that WGAN consistently outperformed the other methods, which supports the utility of WGAN.

7.4 Analysis of Number of Agents

We investigate how accuracy fluctuates according to the number of agents in the context of length 15, as shown in Fig. 5. Overall, as the number of agents increases, the accuracies of all the methods tend to decline. Among them, WGAN achieved the best results in most of the cases. This suggests that WGAN can stably predicts addressees and responses in conversations with many participants.

8 Conclusion and Future Work

We have introduced *multilingual addressee and response selection* by providing (i) formal task definitions, (ii) several knowledge-transfer methods and (iii) a multilingual conversation corpus and dataset. Experimental results have demonstrated that our methods allow models to adapt multiple target languages. In particular, methods for language-invariant models can simultaneously deal with multiple languages with a single model.

Since our methods and dataset can apply to response generation, tackling the multilingual response generation tasks is an interesting line of our future work. In addition, our language-invariant systems can receive conversation in a language (e.g., English) and reply to it in another language (e.g., German). It can lead to interesting findings that our system is evaluated on code-mixing situations, where two or more languages are used in the same context.

References

Rieks Akker and David Traum. 2009. A comparison of addressee detection methods for multiparty conversations. In *Workshop on the Semantics and Pragmatics of Dialogue*.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Dan Bohus and Eric Horvitz. 2011. Multiparty turn taking in situated dialog: Study, lessons, and directions. In *Proceedings of the SIGDIAL 2011 Conference*, SIGDIAL '11, pages 98–109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification.

Kyunghyun Cho, Bart van Merrienboer, aglar Gülehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv: 1408.6988*.

Natasa Jovanović and op den Rieks Akker. 2004. Towards automatic addressee identification in multi-party dialogues. In *Proceedings of SIGDIAL*.

Natasa Jovanović, op den Rieks Akker, and Anton Nijholt. 2006. Addressee identification in face-to-face meetings. In *Proceedings of EACL*.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, Matthew Henderson, and Koichiro Yoshino. 2016. The fifth dialog state tracking challenge. In *Proceeding of Spoken Language Technology Workshop (SLT)*, pages 511–517.

Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, pages 3521–3526.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of ACL*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of EMNLP*, pages 2122–2132.

Ryan Lowe, Nissan Pow, Iulian V. Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of SIGDIAL*, pages 285–294.

Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 264–269, Los Angeles, September. Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of ACL*, pages 1116–1126.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Proceedings of NIPS*, pages 1367–1375.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2017. Coherent dialogue with attention-based language models. In *Proceedings of AAAI*.

Yukiko I. Nakano, Naoya Baba, Hung-Hsuan Huang, and Yuki Hayashi. 2013. Implementation and evaluation of a multimodal addressee identification mechanism for multiparty conversation systems. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 35–42, New York, NY, USA. ACM.

Shuyo Nakatani. 2010. Language detection library for java.

Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *Proceedings of EMNLP*, pages 2133–2143.

Kishore Papineni, Salim E. Roucos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Suman V Ravuri and Andreas Stolcke. 2014. Neural network models for lexical addressee detection. In *Proceedings of INTERSPEECH*, pages 298–302.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of EMNL*, pages 583–593.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of AAAI*, pages 3776–3783.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of ACL/IJCNLP*, pages 1577–1586.

David Traum. 2003. Issues in multiparty dialogues. *Advances in Agent communication*, pages 201–211.

David C Uthus and David W Aha. 2013. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, pages 106–121.

Oriol Vinyals and V. Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv: 1506.05869*.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of EMNLP*, pages 935–945.

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-based deep matching of short texts. In *Proceedings of IJCAI*, pages 1354–1361.

A Hyper-Parameters

Hyper-parameter	Values
Embedding size	512
GRU state size	256
WGAN λ	0.50
WGAN iterations	5
Critic hidden size	512
Critic activation function	ReLU
Batch size	32
Max epoch	30
Adam alpha	{0.001, 0.0005, 0.0001}
L2 weight decay	{0.001, 0.0005, 0.0001}

Table 4: Hyper-parameters for our experiments.

B Statics of Dataset

Tab. 5 shows the details of our dataset. "Docs" is documents, "Utters" is utterances, "W. / U." is the number of words per utterance, "A. / D." is the number of agents per document.

	Dataset (Train / Dev / Test)				
	English (`En`)	Italian (`It`)	Croatian (`Hr`)	German (`De`)	Portuguese (`Pt`)
No. of Docs	7355 (6,606/ 367/ 382)	357 (306/ 17/ 34)	254 (216/ 12/ 26)	248 (216/ 12/ 20)	211 (180/ 10/ 21)
No. of Utters	2.4 M (2.1 M / 13.2 k / 15.1 k)	165 k (144 k / 7 k / 14 k)	80 k (71 k / 3.4 k / 6.9 k)	38 k (33 k / 1.9 / 2.9 k)	52 k (44 k / 2.1 / 6.1 k)
No. of Words	27.0 M (23.8 M/ 1.5 M/ 1.7 M)	1.1 M (1.0 M/ 54 k/ 100 k)	630 k (553 k/ 25 k/ 52 k)	335 k (294 k/ 16 k/ 24 k)	285 k (243 k/ 11 k/ 30 k)
No. of Samples	- 665.6 k/ 45.1 k/ 51.9 k	- 38511 / 2561 / 3873	- 11387 / 512 / 1145	- 5500 / 354 / 569	- 5951 / 285 / 975
Avg. W. / U.	11.1 (11.1/ 11.2/ 11.3)	7.2 (6.9/ 7.7/ 7.1)	7.5 (7.7/ 7.3/ 7.5)	8.5 (8.9/ 8.4/ 8.2)	5.2 (5.5/ 5.2/ 4.9)
Avg. A. / D.	26.8 (26.3/ 30.68/ 32.1)	25.6 (24.9/ 26.2/ 25.6)	12.9 (12.7/ 13.5/ 12.7)	16.4 (17.4/ 15.9/ 15.8)	19.0 (19.7/ 18.6/ 18.8)

Table 5: Statistics of the multilingual dataset.

C Results for each language

Tab. 6 shows the detailed results of single-language adaptation. Tab. 7 shows the detailed results of two-language adaptation. Tab. 8 shows the detailed results of five-language adaptation.

Target	#Train	Method	$\|\mathcal{R}\| = 2$			$\|\mathcal{R}\| = 10$		
			ADR−RES	ADR	RES	ADR−RES	ADR	RES
It	38,511	CHANCE	2.99	5.97	50.00	0.60	5.97	10.00
		TF-IDF	43.89	67.49	64.58	16.63	67.49	23.42
		TRGONLY	63.28	79.86	78.36	32.87	80.92	38.73
		ENONLY	44.54	72.37	60.11	9.91	66.74	16.29
		FINETUNE	64.81	80.79	79.14	34.57	81.44	41.26
		JOINT	63.44	79.81	78.36	34.37	80.30	40.82
		WGAN	65.17	80.56	79.94	35.71	80.76	42.14
Hr	11,387	CHANCE	5.39	10.78	50.00	1.08	10.78	10.00
		TF-IDF	35.63	58.78	61.05	10.22	58.78	17.29
		TRGONLY	40.52	63.06	64.10	14.32	63.23	22.97
		ENONLY	34.06	60.87	54.24	7.95	60.52	12.31
		FINETUNE	40.00	62.62	63.23	14.67	64.72	22.79
		JOINT	44.37	62.97	69.26	15.98	62.97	24.54
		WGAN	45.07	62.62	70.48	16.59	63.76	25.68
De	5,500	CHANCE	4.09	8.17	50.00	0.82	8.17	10.00
		TF-IDF	36.38	64.67	55.36	10.19	64.67	14.41
		TRGONLY	43.94	67.49	64.15	16.52	66.96	22.50
		ENONLY	36.56	63.27	59.75	5.98	57.12	12.13
		FINETUNE	50.44	68.89	72.06	20.39	68.19	26.71
		JOINT	50.79	70.30	70.47	21.79	69.24	27.94
		WGAN	52.90	71.53	72.23	22.50	68.89	28.30
Pt	5,951	CHANCE	3.42	6.84	50.00	0.68	6.84	10.00
		TF-IDF	42.15	68.92	61.44	13.13	68.92	18.87
		TRGONLY	41.64	66.67	62.77	13.95	67.79	20.31
		ENONLY	37.13	66.36	56.51	10.15	65.13	14.26
		FINETUNE	43.08	66.05	64.92	14.97	66.97	21.85
		JOINT	47.59	68.10	69.44	17.13	68.92	24.21
		WGAN	49.54	69.23	70.36	18.56	67.38	25.44
Avg.	-	CHANCE	3.97	7.94	50.00	0.80	7.94	10.00
		TF-IDF	39.51	64.97	60.61	12.54	64.97	18.50
		TRGONLY	47.35	69.27	67.35	19.42	69.73	26.13
		ENONLY	38.07	65.72	57.65	8.50	62.38	13.75
		FINETUNE	49.58	69.59	69.84	21.15	70.33	28.15
		JOINT	51.55	70.30	71.88	22.32	**70.36**	29.38
		WGAN	**53.17**	**70.99**	**73.25**	**23.34**	70.20	**30.39**

Table 6: Results for single-language adaptation. Each number represents accuracy on addressee-response selection (ADR−RES), addressee selection (ADR) or response selection (RES). #Train is the number of training data.

Target	Method	$\|\mathcal{R}\| = 2$ ADR−RES	ADR	RES	$\|\mathcal{R}\| = 10$ ADR−RES	ADR	RES
En, It	CHANCE	1.80 (0.62, 2.95)	3.61	50.00	0.36 (0.12, 0.60)	3.61	10.00
	TF-IDF	40.51 (37.13, 43.89)	61.56	66.24	16.04 (15.44, 16.63)	61.56	26.31
	ENONLY	50.01 (55.47, 44.54)	70.95	69.13	20.59 (31.27, 9.91)	68.05	29.11
	FINETUNE	59.13 (53.44, 64.81)	74.71	77.78	31.07 (27.56, 34.57)	74.62	39.34
	JOINT	59.56 (55.67, 63.44)	74.63	78.50	33.04 (31.71, 34.37)	74.77	41.72
	WGAN	60.20 (55.23, 65.17)	74.94	79.10	33.38 (31.04, 35.71)	75.09	41.87
En, Hr	CHANCE	2.35 (0.62, 5.39)	4.71	50.00	0.47 (0.12, 1.08)	4.71	10.00
	TF-IDF	36.76 (37.13, 35.63)	60.15	61.63	12.82 (15.44, 10.22)	60.15	21.80
	ENONLY	44.77 (55.47, 34.06)	65.20	66.20	19.61 (31.27, 7.95)	64.94	27.12
	FINETUNE	46.03 (52.06, 40.00)	65.29	69.15	20.96 (27.24, 14.67)	66.01	30.28
	JOINT	49.49 (55.66, 43.32)	65.98	73.14	22.95 (31.74, 14.15)	66.15	32.23
	WGAN	49.80 (54.53, 45.07)	65.70	73.96	23.61 (30.62, 16.59)	66.38	33.52
En, De	CHANCE	3.01 (0.62, 4.08)	6.01	50.00	0.60 (0.12, 0.82)	6.01	10.00
	TF-IDF	36.38 (37.13, 36.38)	57.20	64.47	12.83 (15.44, 10.19)	57.20	23.24
	ENONLY	46.02 (55.47, 36.56)	66.40	68.95	18.63 (31.27, 5.98)	63.24	27.03
	FINETUNE	52.22 (53.99, 50.44)	68.79	74.52	24.66 (28.93, 20.39)	68.20	33.09
	JOINT	53.04 (55.28, 50.79)	69.73	74.30	26.09 (31.97, 20.21)	68.98	35.14
	WGAN	54.05 (55.20, 52.90)	70.35	75.19	27.05 (31.42, 22.67)	69.43	35.31
En, Pt	CHANCE	2.02 (0.62, 3.42)	4.04	50.00	0.40 (0.12, 0.68)	4.04	10.00
	TF-IDF	39.64 (37.13, 42.15)	62.27	64.67	14.29 (15.44, 13.13)	62.27	24.03
	ENONLY	46.30 (55.47, 37.13)	67.94	67.33	20.71 (31.27, 10.15)	67.24	28.09
	FINETUNE	46.53 (49.98, 43.08)	66.39	68.97	20.52 (26.06, 14.97)	66.74	29.13
	JOINT	51.39 (55.18, 47.59)	68.66	73.81	24.32 (31.51, 17.13)	69.11	33.28
	WGAN	52.49 (55.44, 49.54)	69.31	74.29	24.88 (31.19, 18.56)	68.16	33.75
Avg.	CHANCE	2.30	4.59	50.00	0.46	4.59	10.00
	TF-IDF	38.32	60.29	64.25	13.99	60.29	23.84
	ENONLY	46.77	67.62	67.90	19.88	65.86	27.83
	FINETUNE	50.98	68.79	72.60	24.30	68.89	32.96
	JOINT	53.37	69.75	74.94	26.60	69.75	35.59
	WGAN	**54.14**	**70.07**	**75.63**	**27.23**	**69.76**	**36.11**

Table 7: Results for two-language adaptation. Each number is macro average of the accuracies over all the languages. Each of the parenthesized numbers in the ADR−RES column is ADR−RES accuracy for each language.

Target	Method	$\|\mathcal{R}\| = 2$ ADR−RES	ADR	RES	$\|\mathcal{R}\| = 10$ ADR−RES	ADR	RES
En, It	TF-IDF	39.04 (37.13, 43.89, 35.63, 36.38, 42.15)	63.10	62.06	13.12 (15.44, 16.63, 10.22, 10.19, 13.13)	63.10	20.64
	ENONLY	41.55 (55.47, 44.54, 34.06, 36.56, 37.13)	66.48	61.75	13.05 (31.27, 9.91, 7.95, 5.98, 10.15)	63.77	19.38
Hr, De, Pt	JOINT	50.69 (54.49, 61.71, 43.41, 47.80, 46.05)	69.00	72.18	22.80 (31.26, 30.29, 14.59, 20.21, 17.64)	69.18	31.11
	WGAN	**52.11** (53.88, 63.18, 44.19, 52.02, 47.28)	**69.74**	**73.34**	**23.39** (30.6, 31.29, 14.67, 22.32, 18.05)	**69.35**	**31.88**

Table 8: Results for five-language adaptation. Each number is macro average of the accuracies over all the languages. Each of the parenthesized numbers in the ADR−RES column is ADR−RES accuracy for each language.