21 Advanced Topics 4: Multi-lingual, Multi-task Models

Up until now, we have assumed that in the case of translation that we would be translating from one particular type of string to another, for example one language to another language in the case of MT. In this section we cover creation of models that work well across a number of languages, a number of tasks, or a number of modalities.

21.1 Methods for Utilizing Heterogeneous Sources

Before delving into the details of the various models that have been proposed, it is worth mentioning several approaches that can take advantage of multiple heterogeneous types of data.

21.1.1 Ensembling

The first method, **ensembling**, consists of combining the prediction of multiple independently trained models together. In the case of learning from multi-lingual or multi-modal data, this means that we will have several models that make predictions based on different types of heterogeneous input. This will be covered more extensively in the materials in Section 18, and thus we will not cover the details here.

21.1.2 Multi-task Learning

The second method, **multi-task learning** [1], is a model training method that attempts to simultaneously learn models for multiple tasks, in the hope that some of the information learned from one of the tasks will be useful in solving the other. This is easiest to understand in the context of neural networks, where the parameters specifying the hidden states allow us to learn compact representations of the salient information required for any particular task. If we perform multi-task learning, and the information needed to solve these two tasks overlap in some way, then training a single model on the two tasks could potentially result in learning better representations overall, increasing the accuracy on both tasks.

The simplest way of doing multi-task learning is to simply define two loss functions that we care about ℓ_1 and ℓ_2, and define our total loss as the sum of these two loss functions. Thus, the total corpus-level loss for a multi-task model will be the sum of the losses over the appropriate training corpora $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively:

$$\ell(\mathcal{C}_1, \mathcal{C}_2) = \ell_1(\mathcal{C}_1) + \ell_2(\mathcal{C}_2). \tag{184}$$

Once we have defined this loss, we can perform training as we normally do through stochastic gradient descent, calculating the loss for each of the tasks and performing parameter update appropriately.

21.1.3 Transfer Learning

The third method, **transfer learning** [27], is also based on learning from data for multiple tasks. Essentially, transfer learning usually consists *transferring* knowledge learned on one task with large amounts of data to another task with smaller amounts of data. This could be viewed as a subset of multi-task learning where we mainly care about the results from only a

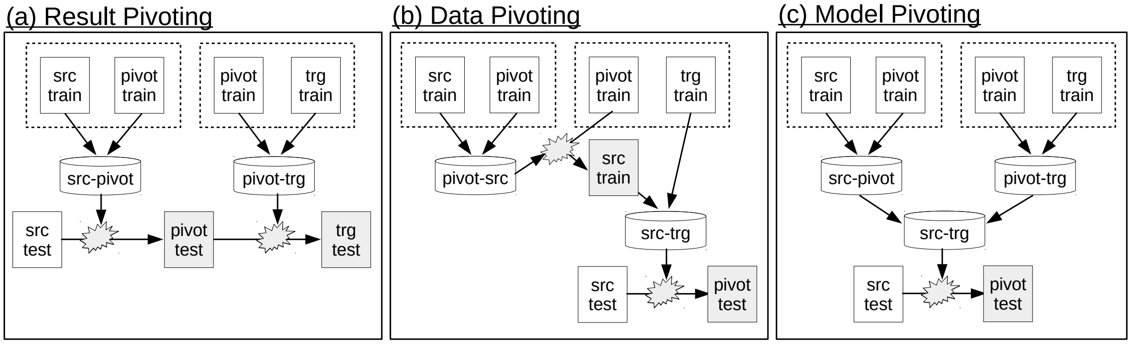

Figure 61: Three varieties of pivoting techniques.

single task The simplest way of doing so is to first train a model on task 1, then after training has concluded, start training on the actual task of interest task 2, which has significantly less training data, although there are many more sophisticated methods.

As a specific subset of transfer learning, we also often hear about **domain adaptation**. This is where we train a model on data from a domain that has a large amount of training data, then attempt to perform model transfer so that the model works well on a target domain with less training data.

21.2 Multi-lingual Models

The first, and perhaps most obvious target for leveraging heterogeneous information sources from the point of view of language is through the use of data from multiple language pairs.

21.2.1 Pivot Translation

One widely used example of practical importance is the case where we want to train a translation system, but have little or no data in the particular language pair. For example, we may want to train a system for Spanish-Japanese translation, and have Spanish-English and English-Japanese translation data, but no direct Spanish-Japanese data. **Pivot translation** is the name for a set of methods that allow us to leverage this data in source-pivot and pivot-target languages to improve translation in our language pair of interest. There are a number of ways to perform pivoting, summarized in Figure 61 and explained in detail below.

Result pivoting: Also called the **direct pivoting** method, this simple method uses existing source-pivot and pivot-target systems to translate our source input to the pivot language, then from the pivot to the target language. Put more formally, if our source sentence is F, our pivot sentence G, and our target sentence E, then this would involve solving the following two equations using our statistical MT systems:

$$\hat{G} = \underset{G}{\text{argmax}}\ P(G \mid F)$$
$$\hat{E} = \underset{E}{\text{argmax}}\ P(E \mid \hat{G})$$

This method is simple and allows for the use of existing systems, but also suffers from error propagation, where mistakes in the pivot output of the first system result in compounding errors in the final output of the second system. These problems can be resolved to some extent by outputting an n-best list from the first system, and then translating each of the n-best hypotheses using the second system, then picking the best final result [28]. However, this results in an n-fold increase in comptuation time for the second translation system, which may not be acceptable in many practical systems.

Data pivoting: A second method for pivoting works at training time by creating **pseudo-parallel data** used to train a translation system in our final language of interest [9]. In the example above, this means that we would first take our source-pivot corpus and use it to train a pivot-source translation system. We then take our pivot-target data, and use this pivot-source system to translate the pivot side into the source language, resulting in a source-target corpus where the source part is machine translated from the pivot language.[54] This data can then be used to directly train a source-target translation system, although it will obviously not be perfect due to the fact that the source data is machine translated, and thus contains errors.

Model pivoting: The final method for pivoting, also called **triangulation**, trains models on the source-pivot and pivot-target pairs, and then combines together the statistics in the model from each language to create a final model [6]. This is easiest to understand from the context of phrase-based machine translation systems, where the source-pivot and pivot-target translation models have phrase translation probabilities $P(\boldsymbol{g} \mid \boldsymbol{f})$ and $P(\boldsymbol{e} \mid \boldsymbol{g})$ respectively. We can then approximate the phrase translation probability between the source and the target by summing over the possible pivot sentences that could be found in the middle:

$$P(\boldsymbol{e} \mid \boldsymbol{f}) \approx \sum_{\boldsymbol{g}} P(\boldsymbol{e} \mid \boldsymbol{g}) P(\boldsymbol{g} \mid \boldsymbol{f}). \tag{185}$$

This approximated probability then can be used as-is in a phrase-based machine translation system instead of the probabilities directly learned from translation data. This model pivoting method has the advantage of not making any hard decisions anywhere in the process, and in the context of symbolic translation models has generally been viewed as the most robust method for making pivoted systems.

21.2.2 Multi-lingual Training

In contrast to the pivoting models in the previous section, which attempted to create models for a particular under-resourced language pair, there are also models that attempt to learn better systems for all languages by sharing training data among various language pairs. Taking the previous example, this would mean that we would want to create better Japanese-English and Spanish-English models by using data from both languages.

Multi-task Learning Approaches: The most straightforward way to do so is through multi-task learning, which has shown promising results particularly for neural machine translation systems. The simplest instantiation of the multi-task learning approach is when we have multiple source languages, and we want to translate into a particular target language. In this case, we assume we have N training corpora $\{\langle \mathcal{F}_1, \mathcal{E}_1 \rangle, \ldots, \langle \mathcal{F}_N, \mathcal{E}_M \rangle\}$, where each $\mathcal{F}_n$ is in a different language (e.g. $\mathcal{F}_1$ is Japanese, $\mathcal{F}_2$ is Spanish in the example above), but $\mathcal{E}_n$

[54] *Question: We could also think of translating the target side of the source-pivot corpus to create a source-target corpus where the target side is machine translated. However, this is less common. Why do you think that is?*

is always in the same language (e.g. English). When training the neural machine translation system, the parameters of the decoder and softmax can be shared over all languages, as the target language is always the same. For the encoder, it is possible to use a different encoder for every language we handle [10, 12], or use a single shared encoder [16, 13]. The shared encoder approach has the advantage that it can share data across all language pairs, but also relies on the strong assumption that the neural network is strong enough to learn how to handle all possible input languages with the same encoder parameters.

It is also possible to relax the assumption that we are handling a single target language, and create a model that can translate into an arbitrary number of languages. In order to do so, because the model parameters are shared between language pairs, it is necessary to make sure that the model knows what language it must be translating into at any particular time. [12] propose to do so by having a separate decoder for each of the target languages, similarly to how we had a separate encoder for each of the input languages. This indicates that if we want to create a system that translates to or from N languages, we will now have N encoders and N decoders, which is significantly better than training separate models for all $N * (N - 1)$ pairs of languages, as would be standard. It is also possible to perform translation into multiple targets using a single for all target languages, as long as we provide some indication of the target language that we would like to be translating into [16, 13]. For example, we can add a special symbol at the beginning of each sentence indicating the target language, so that an input sentence such as "kare wa ringo wo tabeta" would be input into the system as "_ENGLISH_ kare wa ringo wo tabeta" if we wanted to translate into English, or "_SPANISH_ kare wa ringo wo tabeta" if we wanted to translate into Spanish.

One enticing feature of these models is that they may be able to do away for the need with pivoting at all; if we can create a model that translates from an arbitrary number of languages to an arbitrary number of languages, it may be able to translate between languages even if parallel data is lacking. This testing of models on examples that do not exist in their training data is often called **zero-shot learning**, and a number of papers have reported results in this zero-shot scenario [12, 16]. At the time of this writing, results for the zero-shot case are significantly worse tha those of training with standard parallel data, but data-based pivoting [12] or usage of small amounts of parallel training data [16] have been shown to significantly improve results to the point where they are competitive.

Transfer Approaches: [30] report results on transfer learning for low-resource neural machine translation, where we attempt to create a low-resource machine translation system using data from a higher-resourced language. The method works by first training a system with the high resourced data, then re-training *part* of the system with data in the low-resourced language, while freezing the parameters of some parts of the system. In the case where a French-English system was transferred to perform Uzbek-English translation, the authors found that in general freezing the embeddings of the output words while allowing all other parameters to vary achieved the best results.

Ensembling Approaches: One final application of multi-lingual translation can be found in ensembling approaches, which attempt to combine together predictions made from MT systems handling different languages. **Multi-source translation** works by translating sentences in multiple languages to generate a coherent output. This is applicable in situations where identical content is translated into multiple languages (e.g. Wikipedia articles or TED talks), in which case we can use all of the already-translated languages to improve our results on the yet-to-be-translated languages. There are a number of methods for combining multiple languages, including simply combining together the predictions created by bilingual systems on all of the existing source languages using methods such as those described in Section 18 [26], or by specifically devising multi-source model architectures that perform attention over multiple languages at the same time [29]. It is also possible to perform **multi-target translation**, in which predictions in multiple languages are generated at the same time and language models over the results in one language are used to enforce consistency over the other language [25].

21.3 Multi-modal Models

The models described so far have attempted to solve a single task, translation, albeit in a number of different languages. It is also possible to conceive of models that learn from tasks other than translation, or even modalities other than text.

Multi-task Learning Approaches: There are a number of multi-task learning approaches that attempt to improve performance on some sequence-to-sequence learning task by adding in object functions trained over some other variety of data.

Translation with Syntax: It is also possible to perform neural machine translation with an auxiliary loss function of predicting syntactic information (in the form of combinatorial categorical grammar, CCG, tags) [20, 24]. The motivation behind these methods is that forcing the model to predict the syntactic tags should move the model in a direction where it more explicitly captures syntactic information in its hidden representations, improving generalization of the translator to cases where capturing long-range syntax is necessary.

Multiple NLP Tasks: It is also possible to perform multi-task learning over other NLP tasks. For example, [7] perform multi-task learning over various NLP tasks, and [19] do so with a variety of sequence-to-sequence models using memory.

Summarization with Eye Gaze: [17] tackle a summarization task with an auxiliary loss function based on predicting whether a human reader will spend a significant amount of time reading a particular portion of a sentence or not. This method is based on the assumption that readers will spend a longer time looking at salient portions of the sentence, which should be included in a summary.

Ensemble Approaches: There are also methods to incorporate information from multiple sources into translation. One interesting example is translation where in addition to the input sentence we have access to an image that puts the translation into context. This can be framed as a multi-source translation problem, where once source is a textual sentence, and one source is the input image [15]. It is also possible to think of this as a pivoting problem, where we can retrieve similar images and use their descriptions to bias the output of the translation model [14]. This is similarly important in generation of dialog responses, where a dialog agent may have access to some sort of multi-modal context that can help bias its responses [23].

21.4 Multi-domain Models

Finally, one important consideration in creating models is whether the function well in the target domain. For example, a translation system that functions well overall by incorporating large amounts of data from a wide variety of domains may nonetheless perform poorly when asked to translate sentences in a particular sub-domain (e.g. medical or legal documents, informal conversations). The field of **domain adaptation**, a particular instance of transfer learning, attempts to ensure that trained models function well on the particular domain of interest by incorporating both domain-agnostic larger data and domain-specific smaller data.

Data Selection Approaches: One simple but effective way to adapt language models or translation models to a particular domain is to select a subset of data that more closely matches the target domain, and only train the translation or language model on that data. One criterion that has proven effective in the selection of data for language models is the **log-likelihood differential** between a language model trained on the in-domain data and the data trained on general-domain data [22]. Specifically, if we have an in-domain corpus $\mathcal{E}_{\text{in}}$ and general-domain corpus $\mathcal{E}_{\text{gen}}$, then we train two language models $P_{\text{in}}(E)$ and $P_{\text{gen}}(E)$. Then for each sentence in $\mathcal{E}_{\text{gen}}$ we calculate its log-likelihood differential:

$$\text{diff}(E) = \log P_{\text{in}}(E) - \log P_{\text{gen}}(E). \tag{186}$$

This number basically tells us how much more likely the in-domain model thinks the sentence is than the general-domain model, and presumably sentences with higher differentials will be more likely to be similar to the sentences in the target domain. Finally, we select a threshold, and add all sentences in the general-domain corpus that have a differential higher than the threshold. This can also be done in a multi-lingual fashion to consider information on both sides of the translation pair [2], or using neural language models to improve generalization capability [11].

Transfer Approaches: Another way to perform domain adaptation is by training on all data, but giving priority to the training data from inside the domain. There are a large number of approaches to do so, including:

Incremental Training: When using an SGD-style training algorithm, it is possible to first train on the general-domain data, then update the parameters on only the in-domain data [21]. This simple method is nonetheless effective, in that the latter part of training will be performed exclusively on the in-domain data, which allows this data to have a larger effect on the results than the general-domain data.

Domain Labeling: Another simple and popular way to perform domain adaptation is to add a label to the input specifying the domain [8]. This has been incorporated with some success into symbolic models by adding domain-specific features to the log-linear model [5], and to neural MT by adding a special token similar to the tokens used in multi-lingual translation in subsubsection 21.2.2 [18, 4].

Model-level Combination: It is also common to combine models by combining together multiple models, one trained on the general domain, and one trained on a specific domain. This can be done through interpolating multiple models, as mentioned in Section 3, or through various other methods [3].

21.5 Exercise

One possible exercise for this section is to download data from another language pair and add it to the training data of either your neural or symbolic training data. Compare the difference between when multiple source side languages or multiple target-side languages are used.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*, 19:41, 2007.

[2] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–362, 2011.

[3] Arianna Bisazza, Nick Ruiz, Marcello Federico, and FBK-Fondazione Bruno Kessler. Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proceedings of the 2011 International Workshop on Spoken Language Translation (IWSLT)*, pages 136–143, 2011.

[4] Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*, 2017.

[5] Jonathan H Clark, Alon Lavie, and Chris Dyer. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012.

[6] Trevor Cohn and Mirella Lapata. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 728–735, 2007.

[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[8] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 45, 2007.

[9] Adrià De Gispert and Jose B Marino. Catalan-english statistical machine translation without parallel corpus: bridging through spanish. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 65–68, 2006.

[10] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1723–1732, 2015.

[11] Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 678–683, 2013.

[12] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 866–875, 2016.

[13] Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*, 2016.

[14] Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2399–2409, 2016.

[15] Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based multimodal neural machine translation. In *Proceedings of the 1st Conference on Machine Translation (WMT)*, pages 639–645, 2016.

[16] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*, 2016.

[17] Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1528–1533, 2016.

[18] Catherine Kobus, Josep Crego, and Jean Senellart. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140*, 2016.

[19] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285*, 2015.

[20] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[21] Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 2015 International Workshop on Spoken Language Translation (IWSLT)*, 2015.

[22] Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 220–224, 2010.

[23] Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P Spithourakis, and Lucy Vanderwende. Image-grounded conversations: Multimodal context for natural question and response generation. *arXiv preprint arXiv:1701.08251*, 2017.

[24] Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Syntax-aware neural machine translation using ccg. *arXiv preprint arXiv:1702.01147*, 2017.

[25] Graham Neubig, Philip Arthur, and Kevin Duh. Multi-target machine translation with multi-synchronous context-free grammars. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 293–302, 2015.

[26] Franz Josef Och and Hermann Ney. Statistical multi-source translation. pages 253–258, 2001.

[27] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[28] Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 484–491, 2007.

[29] Barret Zoph and Kevin Knight. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 30–34, 2016.

[30] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1575, 2016.