23 Advanced Topics 5: Multi-lingual 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.

23.1 Pivot Translation

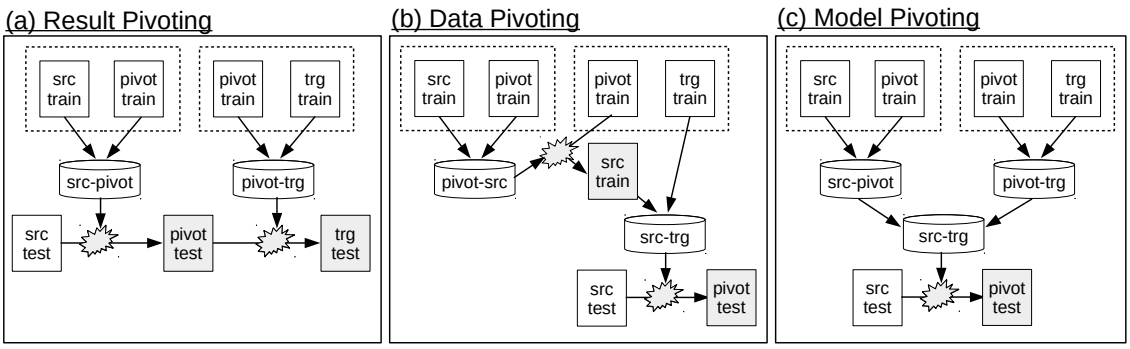

Figure 64: Three varieties of pivoting techniques.

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 64 and explained in detail below.

Result pivoting: Also called **direct pivoting** or **cascade translation**, 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 MT systems:

$$\hat{G} = \operatorname*{argmax}_G P(G \mid F)$$
$$\hat{E} = \operatorname*{argmax}_E P(E \mid \hat{G})$$

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 [15]. 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 [3]. 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.[64] 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.

Within the context of extremely low-resource translation, it may be the case that it is difficult to train a strong pivot-source translation system. However, some work has found that even an extremely simple system such as a word-by-word dictionary replacement system is still sufficient to move the pivot language closer to the source language, and improve results [16].

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 [2]. 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:

$$P(\boldsymbol{e} \mid \boldsymbol{f}) \approx \sum_{\boldsymbol{g}} P(\boldsymbol{e} \mid \boldsymbol{g}) P(\boldsymbol{g} \mid \boldsymbol{f}). \tag{219}$$

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 in the context of phrase-based translation.

23.2 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 $\mathcal{E}_n$

[64] *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 [4, 5], or use a single shared encoder [9, 8]. 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. [5] 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 [9, 8]. 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. In general, multi-lingual translation tasks that use multiple source languages have been more successful than those using multiple target languages, as in the multi-source case the model only needs to learn a single decoder that outputs the target language.

More intelligent parameter sharing methods for many-to-many translation models have also been proposed. For example, it is possible to share the parameters of part of the model but keeping others un-shared [14]. It is even possible to generate parameters for the model on-the-fly for each language under consideration [13]. This can be done by representing the model parameters as a linear interpolation of several basis parameter matrices, where the interpolation coefficients are decided on a language-by-language basis. There have also been the proposal of model architectures that specifically facilitate sharing of the input word embeddings, which are the part of the model that generally are the sparsest and need the most help to be learned properly [6].

One enticing feature of multi-lingual models 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 [5, 9]. 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 [5] or usage of small amounts of parallel training data [9] have been shown to significantly improve results to the point where they are competitive. Another way of improving zero-shot translation results is by bootstrapping them with a pivoted system, training the zero-shot language pairs to match the translations generated by a pivoted system [1].

Transfer Approaches: [18] 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. [11] further expand this to adapting a large multi-lingually trained system to a low resource language. In order to fix problems of overfitting, they use a method of continuing training with some amount of data from other similar languages. [7] examine meta-learning approaches, which attempt to learn a model that is not particularly good on its own, but rather trained specifically to be good when fine-tuned in down-stream tasks.

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 19 [12], or by specifically devising multi-source model architectures that perform attention over multiple languages at the same time [17]. 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 [10].

23.3 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] Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. A teacher-student framework for zero-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1925–1935, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. Universal neural machine translation for extremely low resource languages. 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 344–354. Association for Computational Linguistics, 2018.

[7] Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

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

[9] Melvin Johnson et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*, 2016.

[10] 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.

[11] Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

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

[13] Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, November 2018.

[14] Devendra Sachan and Graham Neubig. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, October 2018.

[15] 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.

[16] Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[17] 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.

[18] 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.