Constraining word alignments with posterior regularization for label transfer

Kevin Martin Jose
Amazon
jskevin@amazon.de

Thomas Gueudre
Amazon
tgueudre@amazon.it

Abstract

Unsupervised word alignments offer a lightweight and interpretable method to transfer labels from high- to low-resource languages, as long as semantically related words have the same label across languages. But such an assumption is often not true in industrial NLP pipelines, where multilingual annotation guidelines are complex and deviate from semantic consistency due to various factors (such as annotation difficulty, conflicting ontology, upcoming feature launches etc.); We address this difficulty by constraining the alignment model to remain consistent with both source and target annotation guidelines, leveraging posterior regularization and labeled examples. We illustrate the overall approach using IBM 2 (fast_align) as a base model, and report results on both internal and external annotated datasets. We measure consistent accuracy improvements on the MultiATIS++ dataset over AWESoME, a popular transformer-based alignment model, in the label projection task (+2.7% at word-level and +15% at sentence-level), and show how even a small amount of target language annotations helps substantially.

1 Introduction

The task of aligning words in parallel sentences (i.e bitexts) originates from statistical machine translation (Brown et al., 1990), where semantic identification was performed based on context similarity in accordance to the well-known *distributional hypothesis*. The most commonly used statistical aligners are built on top of the so-called IBM models (Brown et al., 1993), a series of structured probabilistic models that, while fully unsupervised, often rely on additional assumptions (such as close-to-diagonal alignment) to reach acceptable accuracies. These approaches have since been superseded by neural networks and pretrained embeddings. They nonetheless enjoy a wide popularity across many NLP domains owing to their execution speed, data-efficiency and self-contained implementations.

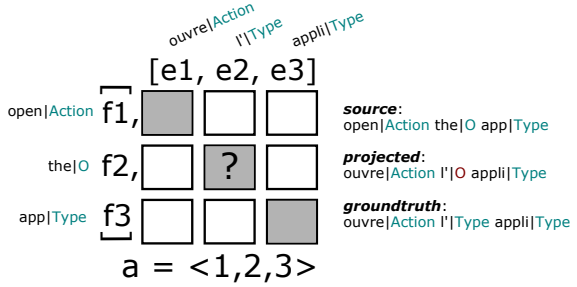

Figure 1: Example of word alignment with notations from English to French. While the identity map is semantically very natural in this example, it conflicts with the ground-truth label. The whole group *l'appli* is labelled as Type in French, possibly to reduce friction with human annotators.

Cheap multilingual word alignments are appealing as they provide a transparent and interpretable way to transfer features from a source language to a target language (see Fig.1). They have been used in the past to transfer costly annotations such as part-of-speech (Yarowsky and Ngai, 2001) or co-reference information from high- to low-resource languages (Postolache et al., 2006). However, the reliability of such a strategy depends on the use case at hand and we argue that it can lead to subtle but systematic failures in downstream tasks. In our industrial use case (that of a voice assistant), multilingual named-entity annotation guidelines factor in a great number of aspects (country launches, available features, human-friendly rules for annotators e.t.c) and end up surprisingly riddled with inconsistencies across languages (see table 1). In such cases, even a slight mismatch between semantics and annotation guidelines will lead to systematic errors: annotation guidelines of the source language "bleed" into the target language. This in turn generates friction for NLP pipelines that rely heavily on annotated resources, such as task oriented dialog systems. In this work, we show how to guide word alignments produced by structured

models to conform to the annotation guidelines of the target language, extending them so that they do not solely rely on semantic relatedness. We use the posterior regularization technique of Ganchev et al. (2010), a general framework that allows integrating information coming from a variety of features as optimization constraints. We illustrate our approach using IBM 2 as the base alignment algorithm. To model the label constraints, we construct n-gram tables that count the frequency of labels assigned to n-grams in the target language. These label n-grams, constructed using the same training data, are then used to bias the alignments so they comply with the annotation scheme. We use an EM-like iterative procedure to train the resulting model - label transfer is done by assigning to targets words the label of their aligned source words.

We evaluate our method on two annotated datasets and show that it combines the strengths of both approaches: the inferred alignments produce better labels than either the baseline aligners or the n-gram models alone. It also remains fast, interpretable, self-contained and data-efficient, which makes it easy to integrate into industrial NLP pipelines. However, it has the same drawbacks that IBM model 2 has (no fertility modelling - i.e cannot handle a single source word generating multiple words in the target language, N-1 source-target mapping, danger of local optima during training). We release our implementation as FastLabel[1].

2 Related Work

Statistical word alignment models continue to be widely used to transfer labels from high- to low-resource languages owing to their speed, low memory footprint and interpretability. Their most famous exponents are the IBM models 1 to 4 (Brown et al., 1993; Och and Ney, 2003), a Bayesian models hierarchy of increasing sophistication. fast_align (Dyer et al., 2013) is a fast reparameterization of IBM Model 2 that significantly cuts down training and inference time. Eflomal (Östling and Tiedemann, 2016) augments IBM model 1 with priors on word order and fertility, and uses Markov Chain Monte Carlo (MCMC) to do inference. Much of the recent work depart from the Bayesian modeling tradition by relying on contextual embeddings to perform the alignment (Pourdamghani et al. 2018, Alkhouli et al. 2018, Sabet et al. 2021). AWESoME (Dou and Neubig, 2021) uses multilingual BERT (Devlin et al., 2019) to extract word alignments, and allows fine-tuning the underlying BERT model on parallel corpora to improve alignment quality. While very accurate, they leverage embeddings from computationally expensive neural networks, and as such, they are not self-contained and the errors made by these models are arguably less interpretable than the simpler statistical models presented here.

Mann and McCallum (2007) introduced expectation regularization as a way to encourage unsupervised model predictions to match an expectation from an external prior. Chang et al. (2007) developed the constraint driven learning (CODL) framework that is capable of allowing different levels of constraint violation. Their formulation, however, did not allow for tractable inference and the authors used beam search to solve the optimization problem. The posterior regularization framework introduced by Ganchev et al. (2010) allows constraint violations while remaining tractable.

Applications of statistical word alignment to label projection are numerous. Label projection using word alignments is discussed in Yarowsky, Ngai, and Wicentowski (2001), Hwa et al. (2005), Östling (2016), Das and Petrov (2011) and Duong et al. (2013). The last three models use the stanford POS tagger (Toutanova et al., 2003) on a high resource source-language and transfer the labels to the target language.

3 Model Formulation

We start with the notations and closely follow (Dyer et al., 2013) for clarity. The source (target) sentence is denoted $\mathbf{f}$ ($\mathbf{e}$), of length n (m). The aim is to infer, from bitexts, an alignment $\boldsymbol{a} = \langle a_1, a_2, \cdots, a_m \rangle$ from source to target: each a_i refers to the position of the source sentence word aligned to the ith word in the target sentence (see Figure 1). We will assume that each target target word is associated to at most one source word: this $N-1$ mapping limitation is not a concern in the context of label projection. In the NER (Named Entity Recognition) setup, both source and target sentences may be annotated with NER labels, and we write $\mathcal{L}$ the set of possible labels, and ℓ_{e_i} (resp. ℓ_{f_j}) the label attached to e_i (resp. f_j) ; $\ell_{\mathbf{e}}$ and $\ell_{\mathbf{f}}$ refer to the label sequences of the whole sentences $\mathbf{e}$ and $\mathbf{f}$.

The parameters of the popular IBM models are usually inferred through maximum likelihood (ML)

[1] https://github.com/amazon-research/fast_label

Dataset	lang	example
MultiATIS++	en	atis_airfare show me round trip fares from denver to philadelphia O O B-round_trip I-round_trip O O B-fromloc.city_name O B-toloc.city_name
	fr	atis_airfare Me montrer les tarifs aller-retour de Denver à Philadelphie O O O O B-round_trip O B-fromloc.city_name O B-toloc.city_name O
	pt	atis_airfare Mostre tarifas de ida e volta de Denver para a Filadélfia O O O B-round_trip I-round_trip I-round_trip O B-fromloc.city_name O O B-toloc.city_name
	de	atis_airfare Zeige mir Tarife für Hin- und Rück flüge von Denver nach Philadelphia O O O O B-round_trip I-round_trip I-round_trip O O B-fromloc.city_name O B-toloc.city_name
	es	atis_airfare Muéstrame las tarifas de ida y vuelta desde Denver hasta Filadelfia O O O O B-round_trip I-round_trip I-round_trip O B-fromloc.city_name O B-toloc.city_name
	zh	atis_airfare 显示从 丹佛 到 费城 的 往返 票价 O B-fromloc.city_name O B-toloc.city_name O B-round_trip O
	hi	atis_airfare डेन्वर से फिलाडेल्फिया के लिए दोतरफा किराए दिखायें B-fromloc.city_name O B-toloc.city_name O O B-round_trip O O
Internal	en	Timer setlo anotherlo timerlaction forlo threellength minuteslength andlo thirtyllength secondslength
	fr	Timer règlelo unlo autrelo minuteurlaction pourlo troislength minuteslength etllength trentellength secondeslength
	en	Weather whatlo today'sldate temperaturedetail
	it	Weather chelo temperaturaldate c'lo èlo oggildate
	en	Appliance turnlaction offlaction thelo boseldevice lightldevice
	pt	Appliance desliguelaction alo luzldevice boseldevice

Table 1: Example training data. The text in teal are word-level labels, and the text in red indicate the overall intent of the sentence. The examples from our internal dataset show some of the discrepancies present in annotation guidelines across languages - for example, the English token-label pair "andlo" corresponds to "etllength" in French. We also observe inconsistencies arising due to word fertility and tokenization choices
- "what" corresponds to "che c' è" (i.e 3 different tokens) in Italian and the two words "turn off" corresponds to the single word "desligue" in Portuguese.

$\theta^* = \arg\max_\theta \mathcal{L}(\theta) = \arg\max_\theta P(\mathbf{e}, \mathbf{f}|\theta)$. The parametric family over which inference is performed depends on the IBM models. In what follows, we illustrate our approach on IBM-2 (as used in fast_align), which comes with a diagonal prior and a set of lexical probabilities representing translations:

$$p_{FA}(e_i, a_i|m, n) = \delta(a_i|i, m, n) \times \theta(e_i|f_{a_i})$$
$$p_{FA}(e_i|m, n) = \sum_{j=0}^{n} p_{FA}(e_i, a_i = j|m, n)$$

where $\delta(\cdot)$ models the diagonal prior and the null alignment probability (Dyer et al., 2013). Because alignments are hidden variables, the ML optimization can only be performed approximately, for example with an Expectation Maximization (EM) iterative scheme. EM can be formulated as an ELBO coordinate ascent (Neal and Hinton, 1998):

$$F(q, \theta) = \log \mathcal{L}(\theta) - D_{KL}(q||p_{FA}(\cdot|\mathbf{e}, \mathbf{f}, m, n))$$
$$\text{E-step} : q^{(t)} = \arg\max_q F(q, \theta^t)$$
$$\text{M-step} : \theta^{(t+1)} = \arg\max_\theta F(q^t, \theta)$$

where q is a reference distribution and is used to inject external knowledge into the optimization, and maximization of the E-step is performed over an arbitrary family of alignments probability distribution. For label projection however, we would like to bias the ELBO optimization so as to favor alignments compatible with the target annotation guidelines, without losing information obtained from the bitexts. The posterior regularization (Ganchev et al., 2010) framework offers an elegant solution, by noting that the E-step above can be easily solved over a constrained set of distributions $\mathcal{Q}$, as long as those constraints are defined in terms of moments of $q \in \mathcal{Q}$:

$$\text{E-step (PR)} : q^{(t)} = \arg\max_{q \in Q} F(q, \theta^t)$$
$$Q = \{q : \mathbb{E}_q [\phi(\mathbf{e}, \mathbf{f}, m, n)] = b\}$$

where ϕ is an arbitrary function. In the context of label projection, we wish to match the projected label distribution $P(\ell_\mathbf{e}|\mathbf{e}, \mathbf{f}, m, n)$ to a reference distribution $r(\ell_\mathbf{e})$, that can be defined quite arbitrarily. Given an alignment a, target words receive the same label as their aligned source words $\ell_{e_i} = \ell_{f_{a_i}} \forall i \in [\mathbf{e}]$. We can therefore rewrite such matching condition as:

$$P(\ell_\mathbf{e}|\mathbf{e}, \mathbf{f}, m, n) = \sum_\mathbf{a} P(\ell_\mathbf{e}|\mathbf{e}, \mathbf{f}, \mathbf{a}) P(\mathbf{a}|\mathbf{e}, \mathbf{f}, m, n) \quad (1)$$

$$= \mathbb{E}_q \left[\mathbb{1} \left(\ell_{\mathbf{e}} = \ell_{f_{\mathbf{a}}} \right) \right] \equiv r(\ell_{\mathbf{e}}),$$

$$\mathbb{1} \left(\ell_{\mathbf{e}} = \ell_{f_{\mathbf{a}}} \right) = \begin{cases} 1, & \text{if } \ell_{\mathbf{e}} = \ell_{f_{\mathbf{a}}} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The set of contraints, one per label configuration per target sentence, is denoted $\mathcal{C}$. In this case, the E-step admits a dual formulation and the optimal alignment distribution q^* has a simple expression in terms for the unconstrained p_{FA}:

$$q^*(\mathbf{a}) = \frac{p_{FA}(\mathbf{a}|\mathbf{e}, \mathbf{f}) e^{-\sum_{c \in \mathcal{C}} \lambda_c^* v_c^{\mathbf{a}}}}{Z(\{\lambda_c^*\})} \quad (3)$$

$$v_c^{\mathbf{a}} = \mathbb{1} \left(\ell_{f_{\mathbf{a}}} = \ell_c \right) - r(\ell_c) \quad (4)$$

$$\lambda_c^* = \arg\max_{\lambda_c} \left[-\log \left(Z(\{\lambda_c^*\}) \right) \right] \forall c \in \mathcal{C} \quad (5)$$

where $\lambda_c, c \in \mathcal{C}$ is a family of Lagrange multipliers enforcing the constraints over label space. The iterative algorithm closely mimics the classical EM coordinate ascent, with the addition of solving the Lagrange multipliers (see Appendix A).

The value of the Lagrange multipliers λ_c^* are computed through gradient ascent over $Z(\{\lambda_c^*\})$. IBM model 2 enjoys the property that its alignment probability p_{FA} factors over the words of each target sentence. It is therefore convenient to split $\mathcal{C}$ accordingly: to each word e_i and each possible $\ell \in \mathcal{L}$, are attached a Lagrange multiplier $\lambda_\ell^{e_i}$ and the cost $v_\ell^{e_i}$ of labelling e_i with ℓ. In such case, $Z(\{\lambda_c^*\})$ further decomposes:

$$Z(\{\lambda_c^*\}) = \prod_{\mathbf{e} \in \text{corp.}} \prod_{e_i \in s} Z_{e_i}(\{\lambda_c^*\})$$

$$Z_{e_i}(\{\lambda_c^*\}) = \sum_{j=1}^{n} p_{FA} \left(a_i = j | \mathbf{e}, \mathbf{f} \right) e^{-\sum_\ell \lambda_\ell^{e_i} v_\ell^{e_i}}$$

$$v_\ell^{e_i} = \mathbb{1} \left(\ell_{f_{a_i}} = \ell \right) - r(\ell)$$

and its derivative w.r.t $\lambda_\ell^{e_i}$:

$$\frac{\partial Z_{e_i}}{\partial \lambda_\ell^{e_i}} = -\sum_{j=1}^{n} p_{FA} \left(a_i = j | e, f \right) v_\ell^{e_i} e^{-\sum_\ell \lambda_\ell^{e_i} v_\ell^{e_i}}$$

The stationary points is reached when $v_\ell^{e_i} = 0$, selecting alignments for which the transferred label distribution matches $r(\ell)$.

4 Experiments

4.1 Baselines

Eflomal[2] and AWESoME[3] were run using the respective authors' publicly released code. The hyperparameter settings used to run these models are described in Appendix C. Since our work is an extension of fast_align, we ported the original fast_align[4] code to Python and extended it to support posterior regularization. Just like the original fast_align implementation, we did 5 iterations of expectation-maximization to train the model. The trained alignment model (i.e q^* in equation 3) is then evaluated on a held out set of bitexts. For each aligned word pair, the label of the source word (usually from an English sentence) is transferred to the aligned target word. All target words aligned to the "null" token are given a label of "o" (for "other"). We then compare the transferred labels to the true labels of the target sentence to calculate the accuracy. Though the label transfer happens at a word level, we report accuracies at the sentence level as well since perfectly annotated sentences are crucial for our industrial use case. The n-gram classifiers in the tables are simple frequency-based classifiers trained on the target language - for a particular n-gram in the test set, the classifier annotates the nth word with the most frequent label assigned to that n-gram in the training data. For n-grams that were not present in the training data (even after backing-off to unigrams), the classifier outputs the label "o" (for "other"). These simple classifiers are essentially the same models that are used to do posterior regularization in our experiments - when used as classifiers, they only output the most likely label for a given n-gram while during regularization we use their entire label distribution.

Lang	Avg. len.	Avg len. of En translation
MultiATIS++		
English (en)	11.05	NA
French (fr)	11.72	11.05 (+6.37%)
Portuguese (pt)	11.96	11.05 (+8.17%)
German (de)	11.29	11.05 (+2.13%)
Spanish (es)	11.88	11.05 (+7.62%)
Chinese (zh)	10.95	11.05 (-1.05%)
Hindi (hi)	10.97	11.05 (-0.73%)
Internal dataset		
Italian (it)	5.29	5.20 (+1.82%)
French (fr)	5.91	5.18 (+14.17%)
Portuguese (pt)	5.42	5.17 (+4.73%)

Table 2: Average sentence lengths (in terms of the number of labelled tokens) for each language present in our datasets. The third column indicates how much longer (or shorter) the sentences in a particular language are compared to their English translations. Unlike MultiATIS++, the English sentences paired with each of languages in our internal dataset are different (i.e the English sentences in the pair en-it are different from those in en-fr), resulting in slightly different average sentence lengths. The translations in both MultiATIS++ and our internal dataset were done by humans.

[2]https://github.com/robertostling/eflomal

[3]https://github.com/neulab/awesome-align

[4]https://github.com/clab/fast_align

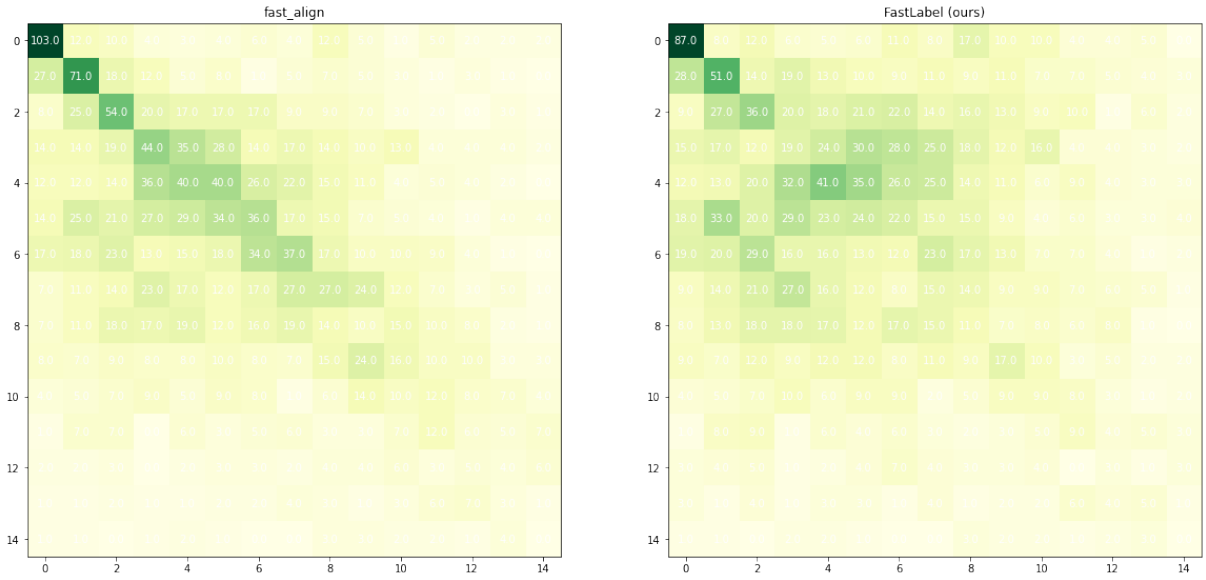

Figure 2: Distribution of word alignments between English-Hindi bitexts in the MultiATIS++ dataset. The left (resp. bottom) axis represents the index of the source (resp. target) word within the source (resp. target) sentence. The left plot shows the distribution of alignments using fast_align. The number inside individual cells represents the frequency of that alignment. The right plot shows the distribution of alignments for the same Engish-Hindi bitexts using FastLabel. We can see from the plots that fast_align has more alignments along the diagonal than FastLabel. Since English and Hindi generally follows different word orders (eg: the Hindi sample present in table 1), the diagonal prior used by fast_align (i.e the assumption that words in target sentence are aligned to the words in relatively the same position in the source sentence) can be problematic. The superior performance of FastLabel (table 3) can be attributed to its ability to overcome fast_align's diagonal prior.

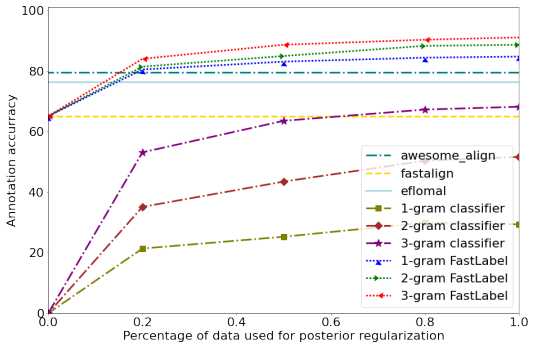

Figure 3: Sentence-level label transfer accuracies between English-German bitexts in MultiATIS++. The amount of German data used to construct the n-gram labels was increased linearly while AWESoMe, eflomal, fast_align, and the word-alignment part of FastLabel were always trained with all available training data.

4.2 Datasets

We ran our experiments on two different datasets - a publicly available corpus of annotated bitexts called MultiATIS++ (Xu et al., 2020) and an internal corpus of annotated bitexts. MultiATIS++ is a multilingual extension of the ATIS (Price, 1990) dataset, which is a transcript of flight information requests to automated airline travel inquiry systems and contains approximately 5000 samples. The queries in ATIS were originally in English and the MultiATIS++ dataset contains annotated human translations of the English queries into six other languages. Our internal dataset consists of queries to a task-oriented dialogue system and contains ten thousand pairs of annotated English-Italian, English-French and English-Portuguese bitexts. The English sentences in the different language pairs in our internal dataset are *not* the same - this means that there is considerable variation in the distribution of intents across different language pairs in this dataset. The scheme for certain type of queries vary across languages (see table 1) as well.

For the set of constraints, we compute a frequency based n-gram model on the annotated monolingual data: the probability of label ℓ_i depends on the word e_i to be labelled, its context of length $n-1$ and the intent of the sentence: $P(\ell_i|\mathbf{e}) = P(\ell_i|e_i, e_{i-1}, \cdots e_{i-n+1}, \textit{intent})$. We include the intent in the counts since labels may strongly depend on it: for example, "play frozen" will be different depending on whether the overall intent is "Music" (resulting in "play|action frozen|album) or "Video" (resulting in "play|action frozen|movie). We construct the n-grams based on the same data that was used to train the word alignment model, and during inference apply the same back-off strategy used by the n-gram classifiers described in the previous section. If an n-gram

	Method	it	fr	pt	de	es	zh	hi
MultiATIS++								
baselines	fast_align	N/A	48.75 (90.44)	40.14 (90.07)	63.821 (94.54)	52.54 (90.54)	43.04 (83.84)	32.31 (85.17)
	eflomal	N/A	67.17 (94.08)	63.56 (93.71)	76.43 (97.20)	66.10 (93.7)	56.58 (87.8)	73.36 (95.00)
	AWESoME	N/A	74.08 (94.94)	73.23 (95.68)	79.59 (97.83)	72.31 (94.95)	55.47 (89.20)	65.06 (94.69)
	1-gram classifier	N/A	27.88 (86.23)	25.51 (84.43)	29.36 (86.68)	29.05 (85.08)	34.38 (81.81)	33.33 (87.44)
	2-gram classifier	N/A	57.88 (93.35)	57.72 (92.79)	59.66 (93.91)	56.79 (92.1)	66.91 (90.56)	59.64 (94.14)
	3- gram classifier	N/A	66.92 (94.91)	67.78 (94.52)	68.21 (95.42)	67.54 (93.43)	67.28 (91.01)	72.80 (96.06)
ours	1-gram FastLabel	N/A	75.81 (95.96)	69.88 (95.84)	84.97 (98.18)	68.92 (94.69)	73.09 (94.11)	76.85 (96.37)
	2-gram FastLabel	N/A	**79.46** (97.10)	78.25 (97.20)	90.53 (98.90)	76.45 (96.39)	**76.43** (**95.13**)	79.03 (97.11)
	3-gram FastLabel	N/A	79.27 (**97.16**)	**78.99** (**97.30**)	**91.09** (**98.96**)	**76.83** (**96.56**)	75.88 (95.11)	**80.34** (**97.23**)
Internal dataset								
baselines	fast_align	x (x')	y (y')	z (z')	N/A	N/A	N/A	N/A
	eflomal	+13.32 (+2.25)	+11.14 (-1.14)	-0.98 (-0.7)	N/A	N/A	N/A	N/A
	AWESoME	+7.49 (+2.07)	+1.4 (+0.02)	+2.4 (+0.55)	N/A	N/A	N/A	N/A
	1-gram classifier	-78.97 (-23.20)	-78.76 (-21.72)	-81.05 (-22.92)	N/A	N/A	N/A	N/A
	2-gram classifier	-76.97 (-22.25)	-76.44 (-20.68)	-78.551 (-22.04)	N/A	N/A	N/A	N/A
	3-gram Classifier	-76.64 (-22.19)	-75.98 (-20.58)	-78.65 (-22.04)	N/A	N/A	N/A	N/A
ours	1-gram FastLabel	+18.48 (+4.36)	+13.62 (+2.89)	+5.08 (+1.48)	N/A	N/A	N/A	N/A
	2-gram FastLabel	**+19.98** (**+4.77**)	**+16.72** (**+3.42**)	**+8.61** (**+2.13**)	N/A	N/A	N/A	N/A
	3-gram FastLabel	+19.65 (+4.67)	+16.10 (**+3.42**)	**+8.61** (**+2.10**)	N/A	N/A	N/A	N/A

Table 3: Percentage of perfectly annotated target sentences obtained as a result of label transfer between bitexts - the word level label transfer accuracy is written inside parentheses. Experiments conducted on our internal dataset report accuracies relative to fast_align.

was not observed in the training data, we leave finding the alignment of the corresponding target word unconstrained. Though we stick to simple frequency-based n-gram models for the sake of speed and interpretability, posterior regularization can accommodate any model that can predict a label distribution, including neural networks.

5 Results

Our results are reported in Table 3. Apart from fast_align, we include eflomal, a more sophisticated statistical alignment model, and AWESoME, a strong model that leverages recent advances in pre-trained language models, as additional baselines. On the MultiATIS++ dataset, FastLabel outperforms AWESoME, our strongest baseline, by around 2.7% at word-level label transfer accuracy and gave around a 15% increase in the amount of perfectly annotated target sentences (averaged across all languages). On our internal dataset, FastLabel resulted in an improvement of around 7% (compared to eflomal, which performed better than AWESoME, averaged across all languages) in the amount of perfectly annotated target sentences. The simple n-gram classifiers perform reasonably well on MultiATIS++. After a deeper inspection, we find that most of the words in this dataset receive the label "O", and entities with richer labels (such as city names) are usually present in both the train and test sets, and makes MultiATIS++ easier to annotate correctly. Our internal dataset is more complex, comprising of 185 intents (eg: "Appliance", "Music") and 211 different label types (i.e "o" or "date" or "song") (for comparison, MultiATIS++ has 23 intents and 122 label types). This is reflected in the much poorer performance of the n-gram classifiers on our internal dataset. Though poor as independent annotators, the same n-gram label distributions are beneficial to FastLabel when used for posterior regularization, indicating that our regularization framework is successful in incorporating the right amount of information from the external prior.

We observe a large drop in performance for fast_align when aligning language from different families (such as English-Chinese bitexts), due to the well-known limitations of the diagonal prior assumption. Moreover, as observed in table 2, Hindi and Chinese sentences are usually slightly shorter than their English counterparts, while the sentences from the other European languages tend to be longer. For example, the Italian translation of the phrase "personalize my echo" could be "personalizza il mio echo" - here the two tokens "my echo" generate three tokens in Italian (high word fertility), while a non-Indo-European language might have the opposite problem with respect to English (low word fertility). Despite these challenges, FastLabel performs comparatively well on these languages thanks to its ability to overcome the diagonal prior of the underlying fast_align algorithm. Figure 2 illustrates the effect of posterior regularization on word-alignments. All subplots show alignments between English-Hindi bitexts in the MultiATIS++ dataset. The plot to the left (fast_align) clearly

id	source	fast_align	ours
1	tracklo alo wetlattribute diaperlitem	enregistrelo unelo couchelattribute culottelattribute mouilléelitem	enregistrelo unelo couchelitem culottelitem mouilléelattribute
2	c.lsource n.lsource n.l source reportlo	lelo comptelsource rendulsource delo c.lsource n.lsource n.lsource	lelo comptelo rendulo delo c.lsource n.lsource n.lsource
3	showlvisual melo anlo octopuslitem	montrelvisual moilother unlo poulpelitem	montrel visual moilvisual unlo poulpelitem

Table 4: Three examples representative of the type of errors in label overcome by posterior regularization. All examples are from the FastLabel evaluated on the English-French bitexts in our internal test dataset. 1) Alignments away from the diagonal - the French word corresponding to "wet" ("mouillée") appear at the end of the sentence. 2) Fertility - "report" is translated into French as "le compte rendu de". 3) Discrepancies in annotation guidelines - though "moi" should be semantically aligned to "me" in the English sentence and hence given the label "o", our internal annotation scheme for French consistently annotates "moi" as "visual" if it follows "montre".

shows a stronger alignment along the diagonal, while this tendency to align along the diagonal is weaker in the plot to the right (FastLabel). Table 4 contains some examples where fast_align made a mistake in transferring the labels from the source sentence, but FastLabel was correct.

How much annotated data is required for FastLabel to improve upon fast_align? Figure 3 reports label transfer accuracy between English-German bitexts in MultiATIS++ using varying amounts of training data to construct the n-gram models. Using only 20% of all available training data to construct the n-gram models gives FastLabel a significant boost over fast_align, demonstrating the applicability of our approach in data-sparse regimes. With growing training data, n-grams become better annotators (to a point where the 3-gram model outperforms fast_align), but a performance gap with FastLabel persists. Although the focus of our work was on maximizing the label transfer accuracy, we also note that posterior regularization resulted in a more semantically accurate translation table (see Appendix B) compared to fast_align.

5.1 Conclusion

We illustrated how to augment existing algorithms (such as fast_align) with information about annotation guidelines, through posterior regularization. Lightweight, self-contained and data-efficient, our approach retains the benefits of statistical aligners while leading to higher quality alignments. It also mitigates semantic inconsistencies that can appear in the annotation guidelines of large scale industrial NLP systems. A natural extension of this work is to use more sophisticated models than n-grams to predict the label distributions. The task of matching the distribution of source labels onto some target through word alignments also bears some similarities with optimal transport. We leave such investigation to the future.

Acknowledgements

We would like to thank Fabian Triefenbach, Markus Boese and Yannick Versley for their feedback on an earlier version of this manuscript.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 280–287, Prague, Czech Republic. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013. Simpler unsupervised pos tagging with bilingual projections. In *ACL*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325.

Gideon S. Mann and Andrew McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 593–600, New York, NY, USA. Association for Computing Machinery.

Radford M. Neal and Geoffrey E. Hinton. 1998. *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer Netherlands, Dordrecht.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling. 2016. A bayesian model for joint word alignment and part-of-speech transfer. In *COLING*.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Oana Postolache, Dan Cristea, and Constantin Orasan. 2006. Transferring coreference chains through word alignment. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Nima Pourdamghani, Marjan Ghazvininejad, and Kevin Knight. 2018. Using word vectors to improve word alignments for low resource machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 524–528, New Orleans, Louisiana. Association for Computational Linguistics.

P. J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2021. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 173–180, USA. Association for Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual nlu.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, page 1–8, USA. Association for Computational Linguistics.

A EM steps with posterior regularization

The iterative algorithm closely mimics the classical EM coordinate ascent, with the addition of solving the Lagrange multipliers:

1. (Start) Random initialization of the IBM 2 model parameters θ_0.
2. Compute p_{FA} as specified by the IBM 2 model, given θ_t.
3. Find the optimal Lagrange multipliers λ_c^* and compute the tilted distribution q^*.
4. Find the optimal parameters θ_{t+1} using q^* in place of p_{FA}.
5. Iterate from step 2 until convergence.

B Excerpt of the translation table for English-French bitexts

fast_align		FastLabel		count
English	French	English	French	
list	courses	list	liste	222
theater	au	theater	theater	22.0
please	te	please	plaît	117.0
closed	est	closed	fermé	5.0
app	application	app	l’	6.0
diaper	culotte	diaper	couche	12.0
don’t	ne	don’t	pas	11.0
march	le	march	mars	32.0
funniest	la	funniest	drôle	3.0
beauty	la	beauty	belle	4.0
cinema	au	cinema	cinéma	9.0
baby	baby	baby	bébé	11.0
mode	mode	mode	multilangues	4.0
frozen	des	frozen	reine	5.0
snow	des	snow	neige	3.0
oatmeal	d’	oatmeal	flocons	3.0
text	un	text	message	3.0
hip	hop	hip	hip	3.0

Table 5: All disagreements appearing more than thrice between the translation tables produced by fast_align and FastLabel on the English-French bitexts in our internal dataset.

In table 5, French words that are not semantic translations of the English source word are highlighted in red. The "count" represents the number of bitexts where the English and French words appeared in the source and target sentences respectively. We observed that posterior regularization using labels improved the quality of the translation table (and consequently, alignments) as well.

C Hyperparameters

Eflomal was run using the "model3" argument so that the final model makes use of IBM model 1, Hidden Markov Models, and also models fertility. Both the forward and reverse alignments (i.e they were not symmetrized) were used to make the priors.

AWESoME was fine-tuned for 2 epochs in an unsupervised fashion independently on the training split of both MultiATIS++ and our internal data, with the following hyperparameters:

hyperparameter	value(s)
extraction	softmax
training epochs	2
training objectives	Masked Language Modelling (MLM), Translation Language Modelling (TLM), Self-training objective (SO)
gradient accumulation steps	4
learning rate	0.00002
maximum training steps	20000

D Compute

FastLabel, eflomal and fast_align were run on cpu on a consumer-grade laptop. AWESoME was fine-tuned for 2 epochs on a single Nvidia Tesla V100 GPU. Our python re-write of fast_align trains at the rate of approximately 260 samples per second. With posterior regularization using trigrams, the training speed drops down to approximately 80 iterations per second. This translates to a training time of 15 seconds per iteration (MultiATIS++ dataset, 4300 training samples) with fast_align and almost 1 minute per training iteration for FastLabel (with trigrams). Though our rewrite of fast_align (and consequently FastLabel) is faster to train compared to recent models such as AWESoME, it is still slower than the original implementation of fast_align and eflomal (which are written in c) - this is currently a limitation of our work and we intend to address this in a future code release.