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 These approaches have since been superseded by neural networks and pretrained embeddings.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.

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 FastLabel1.

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. Eflo-

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

mal (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 (Pour-
damghani et al. 2018, Alkhoulí 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 standard 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 \( f (e) \), of length \( n \) \((m)\). The aim is to infer, from bitexts, an alignment \( a = \langle a_1, a_2, \cdots, a_m \rangle \) from source to target: each \( a_i \) refers to the position of the source sentence word aligned to the \( i \)th 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 \( \ell_{e_i} \) \((\ell_{f_j})\) the label attached to \( e_i \) \((\ell_{f_j})\) ; \( \ell_e \) and \( \ell_f \) refer to the label sequences of the whole sentences \( e \) and \( f \).

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>lang example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiATIS++</td>
<td>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 pt atis_airfare Mostre tarifas de ida e volta de Denver para a Filadélfia O O O B-round_trip I-round_trip O B-fromloc.city_name O B-toloc.city_name es atis_airfare Zeige mir Tarife für Hin- und Rück flüge von Denver nach Philadelphie O O O O B-round_trip I-round_trip O B-fromloc.city_name O B-toloc.city_name de atis_airfare Zeige mir Tarife für Hin- und Rück flüge von Denver nach Philadelphie O O O O B-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 O O O B-round_trip O B-fromloc.city_name O B-toloc.city_name hi atis_airfare डेन्वर से फिलाडेल्फिया के लिए दोहराया किराए दिशायें B-fromloc.city_name O B-toloc.city_name O O B-round_trip O</td>
</tr>
</tbody>
</table>
| Internal      | en Timer setlo anotherlo timeraction forlo threelength minuteslength andlo thirtellength secondlength fr Timer regles lo autrelo minuteurlaction pourlo troisllength minuteslength etlo treillellength secondellength it Weather che lo temperatura date c °o oggi date pt Appliance desligue lo luz device bose device es Appliance desguie lo tiempo device bose device

\( \theta^* = \arg \max_{\theta} \mathcal{L}(\theta) = \arg \max_{\theta} P(e, f | \theta) \) 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):

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

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 \( Q \), as long as those constraints are defined in terms of moments of \( q \in Q \):

\[
\text{E-step (PR) : } q^{(t)} = \arg \max_{q \in Q} F(q, \theta^{(t)})
\]

\[
Q = \{ q : E_q[\phi(e, f, m, n)] = b \}
\]

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

\[
P(\ell_e | e, f, m, n) = \sum_a P(\ell_e | e, f, a) P(a | e, f, m, n)
\]

(1)
The set of constraints, one per label configuration per target sentence, is denoted $C$. In this case, the E-step admits a dual formulation and the optimal alignment distribution $q^*$ has a simple expression in terms of the unconstrained $p_{FA}$:

$$q^*(a) = \frac{p_{FA}(a|e, f) e^{-\sum_{c \in C} \lambda_c^* v_c^a}}{Z(\{\lambda_c^*\})}$$  \hspace{1cm} (3)

$$v_c^a = \mathbb{1} (\ell_f = \ell_c) - r(\ell)$$  \hspace{1cm} (4)

$$\lambda_c^* = \arg \max_{\lambda_c} - \log (Z(\{\lambda_c^*\})) \forall c \in C$$  \hspace{1cm} (5)

where $\lambda_c, c \in C$ 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 $\lambda_c^*$ 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 $C$ accordingly: to each word $e_i$ and each possible $\ell \in L$, are attached a Lagrange multiplier $\lambda_c^\ell_i$ and the cost $v_c^\ell_i$ of labelling $e_i$ with $\ell$. In such case, $Z(\{\lambda_c^\ell_i\})$ further decomposes:

$$Z(\{\lambda_c^\ell_i\}) = \prod_{e \in corp} \prod_{c_i \in s} Z_{e_i}(\{\lambda_c^\ell_i\})$$

$$Z_{e_i}(\{\lambda_c^\ell_i\}) = \sum_{j=1}^n p_{FA} (a_i = j|e, f) e^{-\sum_{c} \lambda_c^\ell_i v_c^{e_i}}$$

$$v_c^{e_i} = \mathbb{1} (\ell_{f_{e_i}} = \ell) - r(\ell)$$

and its derivative w.r.t $\lambda_c^\ell_i$:

$$\frac{\partial Z_{e_i}}{\partial \lambda_c^\ell_i} = - \sum_{j=1}^n p_{FA} (a_i = j|e, f) v_c^{e_i} e^{-\sum_{c} \lambda_c^\ell_i v_c^{e_i}}$$

where $\lambda_c^\ell_i$ are the recordings with which the transferred label distribution matches $r(\ell)$.

4 Experiments

4.1 Baselines

Effomal\textsuperscript{2} and AWESoME\textsuperscript{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\textsuperscript{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 bitexes. 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 backoff 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.

\textsuperscript{2}https://github.com/roberstolleng/effomal

\textsuperscript{3}https://github.com/neulab/awesome-align

\textsuperscript{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 English-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, eftomal, 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 $\ell_i$ depends on the word $e_i$ to be labelled, its context of length $n - 1$ and the intent of the sentence: $P(\ell_i|e) = P(\ell_i|e_i, e_{i-1}, \cdots, e_{i-n+1}, \text{intent})$. 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...
we find that most of the words in this dataset re-
well on MultiATIS++. After a deeper inspection,
word unconstrained. Though we stick to simple
was not observed in the training data, we leave
frequency-based n-gram models for the sake of
was reflected in the much poorer performance of the
can accommodate any model that can predict a
label distributions are beneficial to FastLabel when
5 Results
Our results are reported in Table 3. Apart from
was improved by adding prior information to the
effective at aligning English-Italian parallel senten-
to be longer. For example, the Italian translation of
the well-known limitations of the diagonal prior
be used for posterior regularization, indicating that
our regularization framework is successful in incor-
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 "personal-
alizza 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
because it leverages recent advances in
speed and interpretability, posterior regularization
can accommodate any model that can predict a
label distribution, including neural networks.

### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>it</th>
<th>fr</th>
<th>pt</th>
<th>de</th>
<th>es</th>
<th>zh</th>
<th>hi</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiATIS++</td>
<td>97.16</td>
<td>N/A</td>
<td>76.83</td>
<td>75.88</td>
<td>75.81</td>
<td>76.45</td>
<td>78.25</td>
</tr>
<tr>
<td>ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-gram FastLabel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-gram FastLabel</td>
<td>+19.48</td>
<td>(+4.36)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3-gram FastLabel</td>
<td>+19.65</td>
<td>(+4.67)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>baselines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fast_align</td>
<td>x (x')</td>
<td>y (y')</td>
<td>z (z')</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1-gram classifier</td>
<td>+13.32</td>
<td>(+2.25)</td>
<td>+11.14</td>
<td>(-1.14)</td>
<td>-0.98</td>
<td>(-0.7)</td>
<td>N/A</td>
</tr>
<tr>
<td>2-gram classifier</td>
<td>+7.49</td>
<td>(+2.07)</td>
<td>+1.4</td>
<td>(+0.02)</td>
<td>+2.4</td>
<td>(+0.55)</td>
<td>N/A</td>
</tr>
<tr>
<td>3-gram classifier</td>
<td>-78.97</td>
<td>(-23.20)</td>
<td>-78.76</td>
<td>(-21.72)</td>
<td>-81.05</td>
<td>(-22.92)</td>
<td>N/A</td>
</tr>
<tr>
<td>ours</td>
<td>+19.98</td>
<td>(+3.42)</td>
<td>+16.10</td>
<td>(+3.42)</td>
<td>+8.61</td>
<td>(+2.13)</td>
<td>N/A</td>
</tr>
</tbody>
</table>
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".

<table>
<thead>
<tr>
<th>id</th>
<th>source</th>
<th>fast_align</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>tracklo alo wetattribute diaperitem</td>
<td>enregistréle uno coucheattribute cu-lotlettribute mouilléeitem</td>
<td>enregistréle uno coucheitem cu-lotitem mouilléeattribute</td>
</tr>
<tr>
<td>2</td>
<td>c.source n.source n.l source reportlo</td>
<td>lelo comptelesource rendusource dolo c.source n.source n.source</td>
<td>lelo comptelo rendulolo delo c.source n.source n.source</td>
</tr>
<tr>
<td>3</td>
<td>showvisual melo anto octopusitem</td>
<td>montrovisual moliether unlo montrë visual mouvisual unlo poupleitem</td>
<td>montro visual mouvisual unlo poupleitem</td>
</tr>
</tbody>
</table>

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


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.

A simple, fast, and effective reparameterization of ibm model 2. In NAACL.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. 

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. 

Gideon S. Mann and Andrew McCallum. 2007. 

A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants, pages 355–368. Springer Netherlands, Dordrecht.


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

Robert Östling and Jörg Tiedemann. 2016. 

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).


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. 

End-to-end slot alignment and recognition for cross-lingual nlu.

Inducing multilingual POS taggers and NP brackets via robust projection across aligned corpora. In Second Meeting of the North American Chapter of the 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 $\theta_0$.
2. Compute $p_{FA}$ as specified by the IBM 2 model, given $\theta_t$.
3. Find the optimal Lagrange multipliers $\lambda^*_c$ and compute the tilted distribution $q^*$.
4. Find the optimal parameters $\theta_{t+1}$ using $q^*$ in place of $p_{FA}$.
5. Iterate from step 2 until convergence.

B Excerpt of the translation table for English-French bitexts

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

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

Effomal 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:

<table>
<thead>
<tr>
<th>hyperparameter</th>
<th>value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>extraction</td>
<td>softmax</td>
</tr>
<tr>
<td>training epochs</td>
<td>2</td>
</tr>
<tr>
<td>training objectives</td>
<td>Masked Language Modelling (MLM), Translation Language Modelling (TLM), Self-training objective (SO)</td>
</tr>
<tr>
<td>gradient accumulation steps</td>
<td>4</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.000002</td>
</tr>
<tr>
<td>maximum training steps</td>
<td>20000</td>
</tr>
</tbody>
</table>
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.