Joint part-of-speech and dependency projection from multiple sources

Anders Johannsen   Željko Agić   Anders Søgaard
Center for Language Technology, University of Copenhagen, Denmark
anders@johannsen.com

Abstract

Most previous work on annotation projection has been limited to a subset of Indo-European languages, using only a single source language, and projecting annotation for one task at a time. In contrast, we present an Integer Linear Programming (ILP) algorithm that simultaneously projects annotation for multiple tasks from multiple source languages, relying on parallel corpora available for hundreds of languages. When training POS taggers and dependency parsers on jointly projected POS tags and syntactic dependencies using our algorithm, we obtain better performance than a standard approach on 20/23 languages using one parallel corpus; and 18/27 languages using another.

1 Introduction

Cross-language annotation projection for unsupervised POS tagging and syntactic parsing was introduced fifteen years ago (Yarowsky et al., 2001; Hwa et al., 2005), and the best unsupervised dependency parsers today rely on annotation projection (Rasooli and Collins, 2015).

Despite the maturity of the field, there is an inherent language bias in previous work on cross-language annotation projection. Cross-language annotation projection experiments require training data in m source languages, a parallel corpus of translations from the m source languages into the target language of interest, as well as evaluation data for the target language.[1] Since the canonical resource for parallel text is the Europarl Corpus (Koehn, 2005), which covers languages spoken in the European parliament, annotation projection is typically limited to the subset of Indo-European languages that have treebanks.

Previous work is also limited in another respect. While treebanks typically contain multiple layers of annotation, previous work has focused on projecting data for a single task.

We go significantly beyond previous work in two ways: 1) by considering multi-source projection across languages in parallel corpora that are available for hundreds of languages, including many non-Indo-European languages; and 2) by *jointly* projecting annotation for two mutually dependent tasks, namely POS tagging and dependency parsing. Using multiple source languages makes our projections denser. In single source projection, the source language may not contain all syntactic phenomena of the target language; we combat this by transferring syntactic information from multiple source languages. Our work also differs from previous work on annotation projection in projecting soft rather than hard constraints, i.e., scores rather than labels and edges.

Contributions We present a novel ILP-based algorithm for jointly projecting POS labels and dependency annotations across word-aligned parallel corpora. The performance of our algorithm compares favorably to that of a state-of-the-art projection algorithm, as well as to multi-source delexicalized transfer. Our experiments include between 23 and 27 languages using two parallel corpora that are available for hundreds of languages, namely a collection of Bibles and Watchtower periodicals. Finally, we make both the parallel corpora and the code publicly available.[2]

2 Projection algorithm

The projection algorithm is divided into two distinct steps. First, we *project* potential syntactic

[1] All previous work that we are aware of—with the possible exception of McDonald et al. (2011); but see Sections 2 and 5—uses only a single source ($m = 1$), but in our experiments, we use multiple source languages.

[2] https://bitbucket.org/lowlands/release

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 561–566,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

edges and POS tags from all source languages into an intermediate target graph, which is left deliberately ambiguous. In the second step, we *decode* the target graph by solving a constrained optimisation problem, which simultaneously resolves all ambiguities and produces a single dependency tree with a fixed set of POS tags. Below we describe both steps in more detail.

2.1 Cross-language sentence

The input to our projection algorithm is a *cross-language sentence*, a data structure that ties together a collection of aligned sentences from a parallel corpus, i.e., sentences in many different languages that are determined to be translation equivalents. One sentence of the set is designated as the target while the rest are sources. We project syntactic information from the sources to the target.

All source sentences are automatically parsed with a graph-based dependency parser and labeled with parts of speech. Instead of using the single best dependency tree output by the parser, we extract its scoring matrix, an ambiguous structure that assigns a numeric score to each potential dependency edge. The target sentence is not parsed or POS-tagged. In fact, our approach is explicitly designed to work for target languages where no such resources are available. Only unsupervised word alignments couple the target sentence with each source sentence.

More formally, a cross-language sentence may be represented as a graph $G = (V, E)$, where each vertex is a POS-tagged token of a sentence in some language. With one target and n source languages, the total set of tagged word vertices V can be written as the union of sentence vertices: $V = V_0 \cup \ldots \cup V_n$. The target sentence is $V_t = V_0$, while source sentences are $V_s = V_1 \cup \ldots \cup V_n$.

Two kinds of weighted edges connect the graph. Edges that go between tagged tokens of a sentence V_i represent potential dependency edges. Thus, for the sentence i, the induced subgraph $G[V_i]$ is the (ambiguous) dependency graph. Edges connecting a source vertex to target vertex represent word alignments. The set of alignment edges is $A \subseteq V_s \times V_t$.

To account for POS we introduce a vertex labeling function $l : V \mapsto \Sigma$, where Σ is the POS vocabulary. The source sentences are automatically tagged, and for any source vertex the label function simply returns this tag. For the target sentence the POS labels are unknown, which is to say that every target token is ambiguous between $|\Sigma|$ POS tags. We represent this ambiguity in the graph by creating a vertex for each possible combination of target word and POS. Concretely, if a source sentence i has n tokens, and the target sentence has m tokens, then $|V_i| = n$, and $|V_s| = m|\Sigma|$.

Alignments are constrained such that an alignment $(u, v) \in V_s \times V_t$ only exists if the source and target token were linked by the automatic aligner and $l(u) = l(v)$, i.e., the POS tags match. This filters out potential source relations with dissimilar syntax, a luxury that we are allowed in a multiple source language setup.

2.2 Projecting to ambiguous target graph

The target graph $G[V_t]$ starts out empty and is populated with edges in the following way. We go through the source sentences, looking for potential dependency edges where both endpoints are aligned to the target sentence, and transferring the edge whenever we find one. Technically, for every source sentence i and for each edge in the source graph $(u_s, v_s) \in G[V_i]$, we create an edge (u_t, v_t) in the target iff both (u_s, u_t) and $(v_s, v_t) \in A$. The edge weight is the source edge score (as determined by an automatic parser) weighted by the joint alignment probability of (u_s, u_t) and (v_s, v_t):

$$d(u_t, v_t) = \max_{u_s, v_s} a(u_s, v_s)\ a(u_s, u_t)\ d(v_s, v_t).$$

For clarity, d refers to weights of dependency edges, and a to alignment edge weights. Multiple source sentences may project the same edge to the target graph. When this happens we update the target edge weight only if the new weight is larger than the existing. The weight then reflects the strongest evidence found for a given syntactic relation across all source languages.

2.3 Decoding the target graph

We are now ready to decode the target graph. The result of decoding is a dependency tree as well as a labeling of the target sentences with POS tags. Labeling with POS corresponds to selecting a subset of the vertices $\tilde{V} \subset V_t$, such that exactly one vertex is chosen for each token. Similarly the decoded dependency tree is a subset of the projected target edges with the constraint that it must form a tree over the vertices of $\tilde{V}$. The joint optimization objective is to simultaneously select a set of vertices $\tilde{V}$ and edges $\tilde{E}$ to maximize the score of the decoded tree. We solve this constrained optimization problem by casting it as an integer linear programming (ILP) problem.

The full specification of the ILP model is displayed as Figure 1. The model is optimized over two types of binary decision variables mapping directly to the target graph representation discussed in the previous section, plus additional *flow* variables that enforce tree structure. An edge variable $e_{i,k,j,l}$ represents a target edge (i, j) where the POS of i is k and the POS of j is l. For instance, the variable $e_{2,\mathrm{V},1,\mathrm{N}}$ represents a directed edge from the second token (a verb) to the first (a noun). An active vertex variable $v_{i,k}$ indicates that the POS of token i is chosen as k.

Following Martins (2012), we constrain the search space to spanning trees by using a single-commodity-flow construction. In the commodity-flow analogy, we imagine the root as a factory that produces n commodities (for an n token sentence) which are distributed along the edges of the tree. Each token is a consumer that must receive and pass on all except one commodity to its dependents, i.e., the difference between incoming and outgoing flow should be 1. Since all commodities must be consumed, the outgoing flow for a leaf node will be zero. Together with the requirement that each token must have exactly one head, this ensures all tokens are connected to the root in the tree structure.

The last two constraint groups enforce edge and POS consistency, and the selection of single POS per token. Both are new to this work.

ILP model

$$\begin{array}{rll} \text{Edges} & e_{i,k,j,l} & \in \{0,1\} \\ \text{Vertices} & v_{i,k} & \in \{0,1\} \\ \text{Flow} & \phi_{i,k,j,l} & \in \mathbb{R}^+ \end{array}$$

$$\text{Maximize} \quad \sum_{i,k,j,l} e_{i,k,j,l} \, w_{i,k,j,l}$$

One parent per token

$$\sum_{i,k,l} e_{i,k,j,l} = 1 \qquad \forall j \neq 0$$

The root token (index 0) sends n flow

$$\sum_{j,l} \phi_{0,0,j,l} = n$$

Each token consumes one unit of flow

$$\sum_{i,k,l} \phi_{i,k,x,l} - \sum_{k,j,l} \phi_{x,k,j,l} = 1 \qquad \forall x \neq 0$$

One POS per token

$$\sum_{k} v_{i,k} = 1 \qquad \forall i \neq 0$$

Active edges choose token POS

$$v_{i,k} \geq e_{i,k,j,l} \qquad \forall i \neq 0, j, k, l$$

$$v_{i,l} \geq e_{i,k,j,l} \qquad \forall i, j, k, l$$

Above, i, j, and x are token indices, while k and l refer to POS. Quantification over these symbols in the equations are always with respect to a given target graph.

Figure 1: Specification of the ILP model. We list, in order, the decision variables, the objective, and the five groups of constraint templates.

3 Data sources

Our projection requires parallel text, ideally spanning a large number of languages, and dependency treebanks for the sources.

Treebanks To train the source-side taggers and dependency parsers, and to evaluate the cross-lingual taggers and parsers, we use the Universal Dependencies (UD) version 1.2 treebanks with the corresponding test sets.[3]

Parallel texts We exploit two sources of parallel text: the Edinburgh Multilingual Bible corpus (EBC) (Christodouloupoulos and Steedman, 2014), and our own collection of online texts published by the Wathctower Society (WTC).[4] While the two collections span more than 100 languages, we focus on the subsets that overlap with the UD languages to facilitate evaluation. For EBC, that amounts to 27 languages, and 23 for WTC.

Preprocessing We use simple sentence splitting and tokenization models to segment the parallel corpora.[5] To sentence- and word-align the individual language pairs, we use a Gibbs sampling-based IBM1 alignment model called `efmaral` (Östling, 2015). IBM1 has been shown to lead to more robust alignments across typologically distant language pairs (Östling, 2015). We modify

[3] `http://hdl.handle.net/11234/1-1548`

[4] `https://www.jw.org/`

[5] `https://github.com/bplank/multilingualtokenizer`

the aligner to output alignment probabilities. All the source-side texts are POS-tagged and dependency parsed using TnT (Brants, 2000) and TurboParser (Martins et al., 2013). We use our own fork of the arc-factored TurboParser to output the edge weight matrices.[6]

4 Experiments

4.1 Setup

In our experiments, as in the preprocessing, we use the TnT tagger and the arc-factored TurboParser, which we train on the EBC and WTC texts with projected and decoded annotations. We randomly sample up to 20k sentences per training file in both tagging and parsing. This 20k sampling limit applies to all systems.

We compare two cross-lingual projection-based parsing systems, and one baseline system.

ILP The ILP-based joint projection algorithm we presented in Section 2.

DCA Our implementation of the *de facto* standard annotation projection algorithm of Hwa et al. (2005), as refined by Tiedemann (2014). In contrast to our ILP approach, it uses heuristics to ensure dependency tree constraints on a source-target sentence pair basis. We gather all the pairwise projections into a target sentence graph and then perform maximum spanning tree decoding following Sagae and Lavie (2006).

Delex The multi-source direct delexicalized transfer baseline of McDonald et al. (2011). Each source is represented by an approximately equal number of sentences.

4.2 Results

Table 1 provides a summary of dependency parsing scores. We report UAS scores over predicted and gold POS. The predicted tags come from our cross-lingual taggers. Our ILP approach consistently outperforms DCA on both by a large margin of 3-5 points UAS using predicted POS, and 5-10 points on gold POS. Note that Delex is trained on gold POS and therefore has an advantage in this setting. Relying on predicted POS and WTC data, our ILP approach beats DCA for *all* the test languages. With EBC, we outperform DCA on 19 out of 27 languages.

	Approach		
Predicted POS	ILP	DCA	Delex
EBC	**51.62** (18)	48.39 (8)	42.44 (1)
WTC	**53.58** (20)	48.40 (0)	47.35 (3)
Gold POS			
EBC	**65.43** (25)	59.94 (2)	64.13 (–)
WTC	**66.51** (23)	55.73 (0)	66.68 (–)

Table 1: Macro-averaged UAS scores summarizing our evaluation. EBC: Edinburgh Bible corpus, WTC: Watchtower corpus. Numbers of languages with top performance per system are reported in brackets. All parsers use their respective EBC or WTC taggers.[8]

In Table 2, we split the scores across the test languages and parallel data sources, and we also report the POS tagging accuracies. Our WTC taggers are on average 3.5 points better than EBC taggers, yielding the top score for 16/23 languages from the overlap. Notably, on several non-Indo-European languages, we observe significant improvements. For example, on Indonesian, DCA improves over Delex by 12 points UAS, while ILP adds 6 more points on top. We observe a similar pattern for Arabic and Estonian. We note that Delex tops ILP and DCA on only 1 EBC and 3 WTC languages, and by a narrow margin.

Analysis A projected parse is allowed to be a composite of edges from many source languages. To find out to what degree this actually happens, we analyze all projections into English and German on the WTC corpus.

For German the top four source languages are Czech, Norwegian, French, and English, contributing between 16% and 7% of all edges. For English the top languages are Norwegian, Italian, Indonesian, and Swedish. Here, the top language Norwegian is responsible for 42% of the edges, while Swedish accounts for 13%. Only the language projecting the highest scoring edge is counted. On average, a German sentence has edges from 4.1 source languages. The same number for English is slightly higher, at 4.5.

Manually annotated data We annotate a small number of sentences in English from EBC and

[6] `https://github.com/andersjo/TurboParser`

[8] We do not include Delex in the comparison *for the gold POS scenario only*. In this particular scenario, Delex is also trained on gold POS, and thus biased: the cross-lingual taggers do not have gold POS available for training, and the same holds for Delex and projected POS.

			Dependency parsing					
	POS tagging		EBC			WTC		
Language	EBC	WTC	ILP	DCA	DELEX	ILP	DCA	DELEX
Arabic	39.54	53.91	36.59	37.70	13.17	37.41	32.14	21.15
Basque	43.43	–	22.77	17.38	27.85	–	–	–
Bulgarian	76.45	68.27	50.6	60.03	57.83	49.68	37.18	48.37
Croatian	72.83	76.18	54.19	45.08	42.34	55.16	50.56	45.49
Czech	70.81	78.49	52.67	41.44	40.99	53.09	44.36	47.99
Danish	76.43	86.36	61.14	53.22	49.65	61.78	58.64	55.96
English	71.90	79.2	55.76	50.64	48.04	58.70	57.12	53.87
* Estonian	75.55	73.98	62.9	56.95	49.32	63.85	58.41	48.48
Farsi	64.94	25.67	23.53	42.37	28.93	20.34	12.26	19.48
Finnish	70.41	67.44	43.66	44.51	41.18	42.59	35.6	41.52
French	74.25	79.23	53.52	53.11	48.97	55.69	51.47	51.53
German	74.36	68.36	45.02	50.21	49.36	43.99	36.7	45.79
* Greek	56.52	75.75	62.59	37.73	37.11	62.43	52.95	54.90
Hebrew	43.65	–	30.25	39.76	19.06	–	–	–
Hindi	59.99	48.86	18.26	35.59	21.03	15.95	10.77	21.04
* Hungarian	71.57	71.42	49.74	44.97	43.07	44.17	42.33	46.66
Indonesian	63.30	75.61	51.99	23.53	31.18	58.01	52.29	39.67
Italian	79.28	83.82	63.13	58.66	53.94	64.88	63.57	58.06
* Latin	83.41	–	68.65	68.45	41.42	–	–	–
Norwegian	77.00	85.31	65.04	58.32	53.46	66.54	64.37	60.11
Polish	73.36	73.68	62.94	59.27	53.33	63.74	55.4	54.87
Portuguese	78.41	83.67	63.75	60.45	52.91	64.62	63.16	56.99
* Romanian	71.56	76.34	57.74	56.73	45.73	58.76	54.78	51.23
* Serbian	74.07	–	49.15	49.38	47.06	–	–	–
Slovene	75.68	78.11	59.17	53.66	50.55	59.79	54.8	52.53
Spanish	76.72	85.69	63.63	52.20	47.6	64.93	61.90	55.87
Swedish	78.26	84.80	65.24	55.21	50.85	66.15	62.45	57.48
Average	69.40	73.05	51.62	48.39	42.44	53.58	48.40	47.35
Best for	7	16	18	8	1	20	0	3

Table 2: Tagging and parsing (UAS) accuracy. Scores are macro-averaged, and all parsers use predicted POS from respective EBC or WTC taggers. *: True target languages, not used as sources.

WTC, which gives us a way to directly evaluate the projections without training parsers. On this small test set of 2 × 50 sentences, we obtain UAS scores of 68% (WTC) and 62% (EBC). The POS accuracies are 79% and 80%. All figures are comparable to the results from the indirect projection evaluation.

5 Related work

In recent years, we note an increased interest for work in cross-lingual processing, and particularly in POS tagging and dependency parsing of low-resource languages.

Yarowsky et al. (2001) proposed the idea of inducing NLP tools via parallel corpora. Their contribution started a line of work in annotation projection. Das and Petrov (2011) used graph-based label propagation to yield competitive POS taggers, while Hwa et al. (2005) introduced the projection of dependency trees. Tiedemann (2014) further improved this approach to single-source projection in the context of synthesizing dependency treebanks (Tiedemann and Agić, 2016). The current state of the art in cross-lingual dependency parsing also involves exploiting large parallel corpora (Ma and Xia, 2014; Rasooli and Collins, 2015).

Transferring models by training parsers without lexical features was first introduced by Zeman and Resnik (2008). McDonald et al. (2011) and Søgaard (2011) coupled delexicalization with contributions from multiple sources, while McDonald et al. (2013) were the first to leverage uniform representations of POS and syntactic dependencies in cross-lingual parsing.

Even more recently, Agić et al. (2015) exposed a bias towards closely related Indo-European languages shared by most previous work on annotation projection, while introducing a bias-free projection algorithm for learning 100 POS taggers from multiple sources. Their line of work is non-trivially extended to multilingual dependency parsing by Agić et al. (2016).

The work in annotation projection for cross-lingual NLP invariably treats mutually dependent layers of annotation separately. Our contribution is distinct from these works by implementing the first approach to joint projection of POS and dependencies, while maintaining the outlook on processing truly low-resource languages.

6 Conclusion

In our contribution, we addressed tagging and parsing for low-resource languages through joint cross-lingual projection of POS tags and syntactic dependencies from multiple source languages. Our novel approach to transferring the annotations via word alignments is based on integer linear programming, more specifically on a commodity-flow formalization for spanning trees.

In our experiments with 27 treebanks from the Universal Dependencies (UD) project, our approach compared very favorably to two competitive cross-lingual systems: we provided the best cross-lingual taggers and parsers for 18/27 and 20/23 languages, depending on the parallel corpora used. We made no unrealistic assumptions as to the availability of parallel texts and preprocessing tools for the target languages. Our code and data is freely available.[9]

Acknowledgements This research is partially funded by the ERC Starting Grant LOWLANDS (#313695).

[9] https://bitbucket.org/lowlands/release

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If All You Have is a Bit of the Bible: Learning POS Taggers for Truly Low-Resource Languages. In *ACL*.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual Projection for Parsing Truly Low-Resource Languages. *TACL*, 4.

Thorsten Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. In *ANLP*.

Christos Christodouloupoulos and Mark Steedman. 2014. A Massively Parallel Corpus: The Bible in 100 Languages. *Language Resources and Evaluation*, 49(2).

Dipanjan Das and Slav Petrov. 2011. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *ACL*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*, 11(3).

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit*.

Xuezhe Ma and Fei Xia. 2014. Unsupervised Dependency Parsing with Transferring Distribution via Parallel Guidance and Entropy Regularization. In *ACL*.

André F. T. Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers. In *ACL*.

André F. T. Martins. 2012. *The Geometry of Constrained Structured Prediction: Applications to Inference and Learning of Natural Language Syntax*. Ph.D. thesis, Carnegie Mellon University and Instituto Superior Tecnico.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *EMNLP*.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *ACL*.

Robert Östling. 2015. Word Order Typology through Multilingual Word Alignment. In *ACL*.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-Driven Cross-Lingual Transfer of Dependency Parsers. In *EMNLP*.

Kenji Sagae and Alon Lavie. 2006. Parser Combination by Reparsing. In *NAACL*.

Anders Søgaard. 2011. Data Point Selection for Cross-Language Adaptation of Dependency Parsers. In *ACL*.

Jörg Tiedemann and Željko Agić. 2016. Synthetic Treebanking for Cross-Lingual Dependency Parsing. *Journal of Artificial Intelligence Research*, 55.

Jörg Tiedemann. 2014. Rediscovering Annotation Projection for Cross-Lingual Parser Induction. In *COLING*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. In *NAACL*.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation between Related Languages. In *IJCNLP Workshop on NLP for Less Privileged Languages*.