A Lightweight and High Performance Monolingual Word Aligner

Xuchen Yao and **Benjamin Van Durme**
Johns Hopkins University
Baltimore, MD, USA

Chris Callison-Burch[*]
University of Pennsylvania
Philadelphia, PA, USA

Peter Clark
Vulcan Inc.
Seattle, WA, USA

Abstract

Fast alignment is essential for many natural language tasks. But in the setting of monolingual alignment, previous work has not been able to align more than one sentence pair per second. We describe a discriminatively trained monolingual word aligner that uses a Conditional Random Field to globally decode the best alignment with features drawn from source and target sentences. Using just part-of-speech tags and WordNet as external resources, our aligner gives state-of-the-art result, while being an order-of-magnitude faster than the previous best performing system.

1 Introduction

In statistical machine translation, alignment is typically done as a one-off task during training. However for monolingual tasks, like recognizing textual entailment or question answering, alignment happens repeatedly: once or multiple times per test item. Therefore, the efficiency of the aligner is of utmost importance for monolingual alignment tasks. Monolingual word alignment also has a variety of distinctions than the bilingual case, for example: there is often less training data but more lexical resources available; semantic relatedness may be cued by distributional word similarities; and, both the source and target sentences share the same grammar.

These distinctions suggest a model design that utilizes arbitrary features (to make use of word similarity measure and lexical resources) and exploits deeper sentence structures (especially in the case of major languages where robust parsers are available). In this setting the balance between precision and speed becomes an issue: while we might leverage an extensive NLP pipeline for a language like English, such pipelines can be computationally expensive. One earlier attempt, the MANLI system (MacCartney et al., 2008), used roughly 5GB of lexical resources and took 2 seconds per alignment, making it hard to be deployed and run in large scale. On the other extreme, a simple non-probabilistic Tree Edit Distance (TED) model (c.f. §4.2) is able to align $10,000$ pairs per second when the sentences are pre-parsed, but with significantly reduced performance. Trying to embrace the merits of both worlds, we introduce a discriminative aligner that is able to align tens to hundreds of sentence pairs per second, and needs access only to a POS tagger and WordNet.

This aligner gives state-of-the-art performance on the MSR RTE2 alignment dataset (Brockett, 2007), is faster than previous work, and we release it publicly as the first open-source monolingual word aligner: Jacana.Align.[1]

2 Related Work

The MANLI aligner (MacCartney et al., 2008) was first proposed to align premise and hypothesis sentences for the task of natural language inference. It applies perceptron learning and handles phrase-based alignment of arbitrary phrase lengths. Thadani and McKeown (2011) optimized this model by decoding via Integer Linear Programming (ILP). Benefiting from modern ILP solvers, this led to an order-of-magnitude speedup. With extra syntactic constraints added, the exact alignment match rate for whole sentence pairs was also significantly improved.

Besides the above supervised methods, indirect supervision has also been explored. Among them, Wang and Manning (2010) extended the work of McCallum et al. (2005) and modeled alignment as latent variables. Heilman and Smith (2010) used tree kernels to search for the alignment that

[*] Performed while faculty at Johns Hopkins University.

[1] `http://code.google.com/p/jacana/`

yields the lowest tree edit distance. Other tree or graph matching work for alignment includes that of (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Chambers et al., 2007; Mehdad, 2009; Roth and Frank, 2012).

Finally, feature and model design in monolingual alignment is often inspired by bilingual work, including distortion modeling, phrasal alignment, syntactic constraints, etc (Och and Ney, 2003; DeNero and Klein, 2007; Bansal et al., 2011).

3 The Alignment Model

3.1 Model Design

Our work is heavily influenced by the bilingual alignment literature, especially the discriminative model proposed by Blunsom and Cohn (2006). Given a source sentence $\mathbf{s}$ of length M, and a target sentence $\mathbf{t}$ of length N, the alignment from $\mathbf{s}$ to $\mathbf{t}$ is a sequence of target word indices $\mathbf{a}$, where $a_{m\in[1,M]} \in [0, N]$. We specify that when $a_m = 0$, source word s_t is aligned to a NULL state, i.e., deleted. This models a many-to-one alignment from source to target. Multiple source words can be aligned to the same target word, but not vice versa. One-to-many alignment can be obtained by running the aligner in the other direction. The probability of alignment sequence $\mathbf{a}$ conditioned on both $\mathbf{s}$ and $\mathbf{t}$ is then:

$$p(\mathbf{a} \mid \mathbf{s}, \mathbf{t}) = \frac{\exp(\sum_{m,k} \lambda_k f_k(a_{m-1}, a_m, \mathbf{s}, \mathbf{t}))}{Z(\mathbf{s}, \mathbf{t})}$$

This assumes a first-order Conditional Random Field (Lafferty et al., 2001). The word alignment task is evaluated over F_1. Instead of directly optimizing F_1, we employ softmax-margin training (Gimpel and Smith, 2010) and add a cost function to the normalizing function $Z(\mathbf{s}, \mathbf{t})$ in the denominator, which becomes:

$$\sum_{\hat{\mathbf{a}}} \exp(\sum_{m,k} \lambda_k f_k(\hat{a}_{m-1}, \hat{a}_m, \mathbf{s}, \mathbf{t}) + cost(\mathbf{a_t}, \hat{\mathbf{a}}))$$

where $\mathbf{a_t}$ is the true alignments. $cost(\mathbf{a_t}, \hat{\mathbf{a}})$ can be viewed as special "features" with uniform weights that encourage consistent with true alignments. It is only computed during training in the denominator because $cost(\mathbf{a_t}, \mathbf{a_t}) = 0$ in the numerator. Hamming cost is used in practice.

One distinction of this alignment model compared to other commonly defined CRFs is that the input is two dimensional: at each position m, the model inspects both the entire sequence of source words (as the observation) and target words (whose offset indices are states). The other distinction is that the size of its state space is not fixed (e.g., unlike POS tagging, where states are for instance 45 Penn Treebank tags), but depends on N, the length of target sentence. Thus we can not "memorize" what features are mostly associated with what states. For instance, in the task of tagging mail addresses, a feature of "5 consecutive digits" is highly indicative of a POSTCODE. However, in the alignment model, it does not make sense to design features based on a hard-coded state, say, a feature of "source word lemma matching target word lemma" fires for state index 6.

To avoid this data sparsity problem, all features are defined *implicitly* with respect to the state. For instance:

$$f_k(a_{m-1}, a_m, \mathbf{s}, \mathbf{t}) = \begin{cases} 1 & \text{lemmas match: } s_m, t_{a_m} \\ 0 & \text{otherwise} \end{cases}$$

Thus this feature fires for, e.g.:
$(s_3 = \text{sport}, t_5 = \text{sports}, a_3 = 5)$, and:
$(s_2 = \text{like}, t_{10} = \text{liked}, a_2 = 10)$.

3.2 Feature Design

String Similarity Features include the following similarity measures: Jaro Winkler, Dice Sorensen, Hamming, Jaccard, Levenshtein, NGram overlapping and common prefix matching.[2] Also, two binary features are added for identical match and identical match ignoring case.

POS Tags Features are binary indicators of whether the POS tags of two words match. Also, a "$\text{pos}_{\text{src}}2\text{pos}_{\text{tgt}}$" feature fires for each word pair, with respect to their POS tags. This would capture, e.g., "vbz2nn", when a verb such as *arrests* aligns with a noun such as *custody*.

Positional Feature is a real-valued feature for the positional difference of the source and target word (abs$(\frac{m}{M} - \frac{a_m}{N})$).

WordNet Features indicate whether two words are of the following relations of each other: hypernym, hyponym, synonym, derived form, entailing, causing, members of, have member, substances of, have substances, parts of, have part; or whether

[2] Of these features the trained aligner preferred Dice Sorensen and NGram overlapping.

their lemmas match.[3]

Distortion Features measure how far apart the aligned target words of two consecutive source words are: abs$(a_m + 1 - a_{m-1})$. This learns a general pattern of whether these two target words aligned with two consecutive source words are usually far away from each other, or very close. We also added special features for corner cases where the current word starts or ends the source sentence, or both the previous and current words are deleted (a transition from NULL to NULL).

Contextual Features indicate whether the left or the right neighbor of the source word and aligned target word are identical or similar. This helps especially when aligning functional words, which usually have multiple candidate target functional words to align to and string similarity features cannot help. We also added features for neighboring POS tags matching.

3.3 Symmetrization

To expand from many-to-one alignment to many-to-many, we ran the model in both directions and applied the following symmetrization heuristics (Koehn, 2010): INTERSECTION, UNION, GROW-DIAG-FINAL.

4 Experiments

4.1 Setup

Since no generic off-the-shelf CRF software is designed to handle the special case of dynamic state indices and feature functions (Blunsom and Cohn, 2006), we implemented this aligner model in the Scala programming language, which is fully interoperable with Java. We used the L2 regularizer and LBFGS for optimization. OpenNLP[4] provided the POS tagger and JWNL[5] interfaced with WordNet (Fellbaum, 1998).

To make results directly comparable, we closely followed the setup of MacCartney et al. (2008) and Thadani and McKeown (2011). Training and test data (Brockett, 2007) each contains 800 manually aligned premise and hypothesis pairs from RTE2. Note that the premises contain 29 words on average, and the hypotheses only 11 words. We take the premise as the source and hypothesis as the target, and use S2T to indicate the model aligns from source to target and T2S from target to source.

4.2 Simple Baselines

We additionally used two baseline systems for comparison. One was GIZA++, with the INTERSECTION tricks post-applied, which worked the best among all other symmetrization heuristics. The other was a Tree Edit Distance (TED) model, popularly used in a series of NLP applications (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Heilman and Smith, 2010). We used uniform cost for deletion, insertion and substitutions, and applied a dynamic program algorithm (Zhang and Shasha, 1989) to decode the tree edit sequence with the minimal cost, based on the Stanford dependency tree (De Marneffe and Manning, 2008). This non-probabilistic approach turned out to be extremely fast, processing about 10,000 sentence pairs per second with pre-parsed trees, performing quantitatively better than the Stanford RTE aligner (Chambers et al., 2007).

4.3 MANLI Baselines

MANLI was first developed by MacCartney et al. (2008), and then improved by Thadani and McKeown (2011) with faster and exact decoding via ILP. There are four versions to be compared here:

MANLI the original version.

MANLI-approx. re-implemented version by Thadani and McKeown (2011).

MANLI-exact decoding via ILP solvers.

MANLI-constraint MANLI-exact with hard syntactic constraints, mainly on common "light" words (determiners, prepositions, etc.) attachment to boost exact match rate.

4.4 Results

Following Thadani and McKeown (2011), performance is evaluated by macro-averaged precision, recall, F_1 of aligned token pairs, and exact (perfect) match rate for a whole pair, shown in Table 1. As our baselines, GIZA++ (with alignment intersection of two directions) and TED are on par with previously reported results using the Stanford RTE aligner. The MANLI-family of systems provide stronger baselines, notably MANLI-constraint, which has the best F_1 and exact match rate among themselves.

We ran our aligner in two directions: S2T and T2S, then merged the results with INTERSECTION, UNION and GROW-DIAG-FINAL. Our system beats

[3] We found that each word has to be POS tagged to get an accurate relation, otherwise this feature will not help.

[4] http://opennlp.apache.org/

[5] http://jwordnet.sf.net/

System	P %	R %	F_1 %	E %
GIZA++, ∩	82.5	74.4	78.3	14.0
TED	80.6	79.0	79.8	13.5
Stanford RTE∗	82.7	75.8	79.1	-
MANLI∗	85.4	85.3	85.3	21.3
MANLI-approx.◁	87.2	86.3	86.7	24.5
MANLI-exact◁	87.2	86.1	86.8	24.8
MANLI-constraint◁	89.5	86.2	87.8	33.0
this work, S2T	91.8	83.4	87.4	25.9
this work, T2S	93.7	84.0	**88.6**	**35.3**
S2T ∩ T2S	**95.4**	80.8	87.5	31.3
S2T ∪ T2S	90.3	**86.6**	88.4	29.6
GROW-DIAG-FINAL	94.4	81.8	87.6	30.8

Table 1: Results on the 800 pairs of test data. E% stands for exact (perfect) match rate. Systems marked with ∗ are reported by MacCartney et al. (2008), with ◁ by Thadani and McKeown (2011).

the weak and strong baselines[6] in all measures except recall. Some patterns are very clearly shown:

Higher precision, lower recall is due to the higher-quality and lower-coverage of WordNet, where the MANLI-family systems used additional, automatically derived lexical resources.

Imbalance of exact match rate between S2T and T2S with a difference of 9.4% is due to the many-to-one nature of the aligner. When aligning from source (longer) to target (shorter), multiple source words can align to the same target word. This is not desirable since multiple duplicate "light" words are aligned to the same "light" word in the target, which breaks perfect match. When aligning T2S, this problem goes away: the shorter target sentence contains less duplicate words, and in most cases there is an one-to-one mapping.

MT heuristics help, with INTERSECTION and UNION respectively improving precision and recall.

4.5 Runtime Test

Table 2 shows the runtime comparison. Since the RTE2 corpus is imbalanced, with premise length (words) of 29 and hypothesis length of 11, we also compare on the corpus of FUSION (McKeown et al., 2010), with both sentences in a pair averaging 27. MANLI-approx. is the slowest, with quadratic growth in the number of edits with sentence length. MANLI-exact is in second place, relying on the ILP solver. This work has a precise $O(MN^2)$ decoding time, with M the source sentence length and N the target sentence length.

corpus	sent. pair length	MANLI-approx.	MANLI-exact	this work
RTE2	29/11	1.67	0.08	0.025
FUSION	27/27	61.96	2.45	0.096

Table 2: Alignment runtime in seconds per sentence pair on two corpora: RTE2 (Cohn et al., 2008) and FUSION (McKeown et al., 2010). The MANLI-* results are from Thadani and McKeown (2011), on a Xeon 2.0GHz with 6MB Cache. The runtime for this work takes the longest timing from S2T and T2S, on a Xeon 2.2GHz with 4MB cache (the closest we can find to match their hardware). Horizontally in a real-world application where sentences have similar length, this work is roughly 20x faster (0.096 vs. 2.45). Vertically, the decoding time for our work increases less dramatically when sentence length increases (0.025→0.096 vs. 0.08→2.45).

features	P %	R %	F1 %	E %
full (T2S)	93.7	84.0	88.6	35.3
- POS	93.2	83.5	88.1	31.4
- WordNet	93.2	83.7	88.2	33.5
- both	93.1	83.2	87.8	30.1

Table 3: Performance without POS and/or WordNet features.

While MANLI-exact is about twenty-fold faster than MANLI-approx., our aligner is at least another twenty-fold faster than MANLI-exact when the sentences are longer and balanced. We also benefit from shallower pre-processing (no parsing) and can store all resources in main memory.[7]

4.6 Ablation Test

Since WordNet and the POS tagger is the only used external resource, we removed them[8] from the feature sets and reported performance in Table 3. This somehow reflects how the model would perform for a language without a suitable POS tagger, or more commonly, WordNet in that language. At this time, the model falls back to relying on string similarities, distortion, positional and contextual features, which are almost language-independent. A loss of less than 1% in F_1 suggests that the aligner can still run reasonably well without a POS tagger and WordNet.

[6] Unfortunately both MacCartney and Thadani no longer have their original output files (personal communication), so we cannot run a significance test against their result.

[7] WordNet (~30MB) is a smaller footprint than the 5GB of external resources used by MANLI.

[8] per request of reviewers. Note that WordNet is less precise without a POS tagger. When we removed the POS tagger, we enumerated all POS tags for a word to find its hypernym/synonym/... synsets.

4.7 Error Analysis

There were three primary categories of error:[9]

1. Token-based paraphrases that are not covered by WordNet, such as *program* and *software*, *business* and *venture*. This calls for broader-coverage paraphrase resources.

2. Words that are semantically related but not exactly paraphrases, such as *married* and *wife*, *beat* and *victory*. This calls for resources of close distributional similarity.

3. *Phrases* of the above kinds, such as *elected* and *won a seat*, *politician* and *presidential candidate*. This calls for further work on phrase-based alignment.[10]

There is a trade-off using WordNet vs. larger, noisier resources in exchange of higher precision vs. recall and memory/disk allocation. We think this is an application-specific decision; other resources could be easily incorporated into our model, which we may explore in the future to explore the trade-off in addressing items 1 and 2.

5 Conclusion

We presented a model for monolingual sentence alignment that gives state-of-the-art performance, and is significantly faster than prior work. We release our implementation as the first open-source monolingual aligner, which we hope to be of benefit to other researchers in the rapidly expanding area of natural language inference.

Acknowledgement

We thank Vulcan Inc. for funding this work. We also thank Jason Smith, Travis Wolfe, Frank Ferraro for various discussion, suggestion, comments and the three anonymous reviewers.

References

Mohit Bansal, Chris Quirk, and Robert Moore. 2011. Gappy phrasal alignment by agreement. In *Proceedings of ACL*, Portland, Oregon, June.

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of ACL2006*, pages 65–72.

Chris Brockett. 2007. Aligning the RTE 2006 corpus. Technical report, Microsoft Research.

N. Chambers, D. Cer, T. Grenager, D. Hall, C. Kiddon, B. MacCartney, M.C. de Marneffe, D. Ramage, E. Yeh, and C.D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Comput. Linguist.*, 34(4):597–614, December.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of ACL2007*.

C. Fellbaum. 1998. *WordNet: An Electronical Lexical Database*.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin crfs: training log-linear models with cost functions. In *NAACL 2010*, pages 733–736.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of NAACL 2010*, pages 1011–1019, Los Angeles, California, June.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In *PASCAL Challenges on RTE*, pages 17–20.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA.

B. MacCartney, M. Galley, and C.D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP2008*, pages 802–811.

Andrew McCallum, Kedar Bellare, and Fernando Pereira. 2005. A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, July.

[9] We submitted a browser in JavaScript (AlignmentBrowser.html) in the supporting material that compares the gold alignment and test output; readers are encouraged to try it out.

[10] Note that MacCartney et al. (2008) showed that in the MANLI system setting phrase size to larger than one there was only a 0.2% gain in F_1, while the complexity became much larger.

Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *ACL2010 short*, pages 317–320.

Y. Mehdad. 2009. Automatic cost estimation for tree edit distance using particle swarm optimization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 289–292.

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

Vasin Punyakanok, Dan Roth, and Wen T. Yih. 2004. Mapping Dependencies Trees: An Application to Question Answerin. In *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida.

Michael Roth and Anette Frank. 2012. Aligning predicates across monolingual comparable texts using graph-based clustering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 171–182, Jeju Island, Korea, July.

Kapil Thadani and Kathleen McKeown. 2011. Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *Proceedings of ACL short*.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1164–1172, Stroudsburg, PA, USA.

K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, December.