Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment

Kapil Thadani and **Kathleen McKeown**
Department of Computer Science
Columbia University
New York, NY 10027, USA
{kapil,kathy}@cs.columbia.edu

Abstract

The task of aligning corresponding phrases across two related sentences is an important component of approaches for natural language problems such as textual inference, paraphrase detection and text-to-text generation. In this work, we examine a state-of-the-art structured prediction model for the alignment task which uses a phrase-based representation and is forced to decode alignments using an approximate search approach. We propose instead a straightforward exact decoding technique based on integer linear programming that yields order-of-magnitude improvements in decoding speed. This ILP-based decoding strategy permits us to consider syntactically-informed constraints on alignments which significantly increase the precision of the model.

1 Introduction

Natural language processing problems frequently involve scenarios in which a pair or group of related sentences need to be aligned to each other, establishing links between their common words or phrases. For instance, most approaches for natural language inference (NLI) rely on alignment techniques to establish the overlap between the given premise and a hypothesis before determining if the former entails the latter. Such monolingual alignment techniques are also frequently employed in systems for paraphrase generation, multi-document summarization, sentence fusion and question answering.

Previous work (MacCartney et al., 2008) has presented a phrase-based monolingual aligner for NLI (MANLI) that has been shown to significantly outperform a token-based NLI aligner (Chambers et al., 2007) as well as popular alignment techniques borrowed from machine translation (Och and Ney, 2003; Liang et al., 2006). However, MANLI's use of a phrase-based alignment representation appears to pose a challenge to the decoding task, i.e. the task of recovering the highest-scoring alignment under some parameters. Consequently, MacCartney et al. (2008) employ a stochastic search algorithm to decode alignments approximately while remaining consistent with regard to phrase segmentation.

In this paper, we propose an exact decoding technique for MANLI that retrieves the globally optimal alignment for a sentence pair given some parameters. Our approach is based on integer linear programming (ILP) and can leverage optimized general-purpose LP solvers to recover exact solutions. This strategy boosts decoding speed by an order of magnitude over stochastic search in our experiments. Additionally, we introduce hard syntactic constraints on alignments produced by the model, yielding better precision and a large increase in the number of perfect alignments produced over our evaluation corpus.

2 Related Work

Alignment is an integral part of statistical MT (Vogel et al., 1996; Och and Ney, 2003; Liang et al., 2006) but the task is often substantively different from monolingual alignment, which poses unique challenges depending on the application (MacCartney et al., 2008). Outside of NLI, prior research has also explored the task of monolingual word align-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 254–259,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

ment using extensions of statistical MT (Quirk et al., 2004) and multi-sequence alignment (Barzilay and Lee, 2002).

ILP has been used extensively for applications ranging from text-to-text generation (Clarke and Lapata, 2008; Filippova and Strube, 2008; Woodsend et al., 2010) to dependency parsing (Martins et al., 2009). It has also been recently employed for finding phrase-based MT alignments (DeNero and Klein, 2008) in a manner similar to this work; however, we further build upon this model through syntactic constraints on the words participating in alignments.

3 The MANLI Aligner

Our alignment system is structured identically to MANLI (MacCartney et al., 2008) and uses the same phrase-based alignment representation. An alignment E between two fragments of text T_1 and T_2 is represented by a set of edits $\{e_1, e_2, \ldots\}$, each belonging to one of the following types:

- INS and DEL edits covering unaligned words in T_1 and T_2 respectively
- SUB and EQ edits connecting a phrase in T_1 to a phrase in T_2. EQ edits are a specific case of SUB edits that denote a word/lemma match; we refer to both types as SUB edits in this paper.

Every token in T_1 and T_2 participates in exactly one edit. While alignments are one-to-one at the phrase level, a phrase-based representation effectively permits many-to-many alignments at the token level. This enables the aligner to properly link paraphrases such as *death penalty* and *capital punishment* by exploiting lexical resources.

3.1 Dataset

MANLI was trained and evaluated on a corpus of human-generated alignment annotations produced by Microsoft Research (Brockett, 2007) for inference problems from the second Recognizing Textual Entailment (RTE2) challenge (Bar-Haim et al., 2006). The corpus consists of a development set and test set that both feature 800 inference problems, each of which consists of a premise, a hypothesis and three independently-annotated human alignments. In our experiments, we merge the annotations using majority rule in the same manner as MacCartney et al. (2008).

3.2 Features

A MANLI alignment is scored as a sum of weighted feature values over the edits that it contains. Features encode the type of edit, the size of the phrases involved in SUB edits, whether the phrases are constituents and their similarity (determined by leveraging various lexical resources). Additionally, contextual features note the similarity of neighboring words and the relative positions of phrases while a positional distortion feature accounts for the difference between the relative positions of SUB edit phrases in their respective sentences.

Our implementation uses the same set of features as MacCartney et al. (2008) with some minor changes: we use a shallow parser (Daumé and Marcu, 2005) for detecting constituents and employ only string similarity and WordNet for determining semantic relatedness, forgoing NomBank and the distributional similarity resources used in the original MANLI implementation.

3.3 Parameter Inference

Feature weights are learned using the averaged structured perceptron algorithm (Collins, 2002), an intuitive structured prediction technique. We deviate from MacCartney et al. (2008) and do not introduce L2 normalization of weights during learning as this could have an unpredictable effect on the averaged parameters. For efficiency reasons, we parallelize the training procedure using iterative parameter mixing (McDonald et al., 2010) in our experiments.

3.4 Decoding

The decoding problem is that of finding the highest-scoring alignment under some parameter values for the model. MANLI's phrase-based representation makes decoding more complex because the segmentation of T_1 and T_2 into phrases is not known beforehand. Every pair of phrases considered for inclusion in an alignment must adhere to some consistent segmentation so that overlapping edits and uncovered words are avoided.

Consequently, the decoding problem cannot be factored into a number of independent decisions and MANLI searches for a good alignment using a stochastic simulated annealing strategy. While seemingly quite effective at avoiding local maxima,

System	Data	$P\%$	$R\%$	$F_1\%$	$E\%$
MANLI	dev	83.4	85.5	84.4	21.7
(reported 2008)	test	85.4	85.3	85.3	21.3
MANLI	dev	85.7	84.8	85.0	23.8
(reimplemented)	test	87.2	86.3	86.7	24.5
MANLI-Exact	dev	85.7	84.7	85.2	24.6
(this work)	test	87.8	86.1	86.8	24.8

Table 1: Performance of aligners in terms of precision, recall, F-measure and number of perfect alignments ($E\%$).

Corpus		Size	Approximate Search	Exact ILP
RTE2	dev	800	2.58	0.11
	test	800	1.67	0.08
McKeown et al. (2010)		297	61.96	2.45

Table 2: Approximate running time per decoding task in seconds for the search-based approximate decoder and the ILP-based exact decoder on various corpora (see text for details).

this iterative search strategy is computationally expensive and moreover is not guaranteed to return the highest-scoring alignment under the parameters.

4 Exact Decoding via ILP

Instead of resorting to approximate solutions, we can simply reformulate the decoding problem as the optimization of a linear objective function with linear constraints, which can be solved by well-studied algorithms using off-the-shelf solvers[1]. We first define boolean indicator variables x_e for every possible edit e between T_1 and T_2 that indicate whether e is present in the alignment or not. The linear objective that maximizes the score of edits for a given parameter vector $\mathbf{w}$ is expressed as follows:

$$\begin{aligned} f(\mathbf{w}) &= \max \sum_e x_e \times \mathit{score}_{\mathbf{w}}(e) \\ &= \max \sum_e x_e \times \mathbf{w} \cdot \Phi(e) \qquad (1) \end{aligned}$$

where $\Phi(e)$ is the feature vector over an edit. This expresses the score of an alignment as the sum of scores of edits that are present in it, i.e., edits e that have $x_e = 1$.

In order to address the phrase segmentation issue discussed in §3.4, we merely need to add linear constraints ensuring that every token participates in exactly one edit. Introducing the notation $e \prec t$ to indicate that edit e covers token t in one of its phrases, this constraint can be encoded as:

$$\sum_{e:\, e \prec t} x_e = 1 \qquad \forall t \in T_i,\, i = \{1, 2\}$$

On solving this integer program, the values of the variables x_e indicate which edits are present in the highest-scoring alignment under $\mathbf{w}$. A similar approach is employed by DeNero and Klein (2008) for finding optimal phrase-based alignments for MT.

4.1 Alignment experiments

For evaluation purposes, we compare the performance of approximate search decoding against exact ILP-based decoding on a reimplementation of MANLI as described in §3. All models are trained on the development section of the Microsoft Research RTE2 alignment corpus (cf. §3.1) using the training parameters specified in MacCartney et al. (2008). Aligner performance is determined by counting aligned token pairs per problem and macro-averaging over all problems. The results are shown in Table 1.

We first observe that our reimplemented version of MANLI improves over the results reported in MacCartney et al. (2008), gaining 2% in precision, 1% in recall and 2-3% in the fraction of alignments that exactly matched human annotations. We attribute at least some part of this gain to our modified parameter inference (cf. §3.3) which avoids normalizing the structured perceptron weights and instead adheres closely to the algorithm of Collins (2002).

Although exact decoding improves alignment performance over the approximate search approach, the gain is marginal and not significant. This seems to indicate that the simulated annealing search strategy is fairly effective at avoiding local maxima and finding the highest-scoring alignments.

4.2 Runtime experiments

Table 2 contains the results from timing alignment tasks over various corpora on the same machine using the models trained as per §4.1. We observe a

[1] We use LPsolve: `http://lpsolve.sourceforge.net/`

twenty-fold improvement in performance with ILP-based decoding. It is important to note that the specific implementations being compared[2] may be responsible for the relative speed of decoding.

The short hypotheses featured in the RTE2 corpus (averaging 11 words) dampen the effect of the quadratic growth in number of edits with sentence length. For this reason, we also run the aligners on a corpus of 297 related sentence pairs which don't have a particular disparity in sentence lengths (McKeown et al., 2010). The large difference in decoding time illustrates the scaling limitations of the search-based decoder.

5 Syntactically-Informed Constraints

The use of an integer program for decoding provides us with a convenient mechanism to prevent common alignment errors by introducting additional constraints on edits. For example, function words such as determiners and prepositions are often misaligned just because they occur frequently in many different contexts. Although MANLI makes use of contextual features which consider the similarity of neighboring words around phrase pairs, out-of-context alignments of function words often appear in the output. We address this issue by adding constraints to the integer program from §4 that look at the syntactic structure of T_1 and T_2 and prevent matching function words from appearing in an alignment unless they are syntactically linked with other words that are aligned.

To enforce token-based constraints, we define boolean indicator variables y_t for each token t in text snippets T_1 and T_2 that indicate whether t is involved in a SUB edit or not. The following constraint ensures that $y_t = 1$ if and only if it is covered by a SUB edit that is present in the alignment.

$$y_t - \sum_{\substack{e:\, e \prec t,\\ e \text{ is SUB}}} x_e = 0 \qquad \forall t \in T_i,\ i = \{1, 2\}$$

We refer to tokens t with $y_t = 1$ as being *active* in the alignment. Constraints can now be applied over any token with specific part-of-speech (POS) tag in order to ensure that it can only be active if a different token related to it in a dependency parse of the sentence is also active. We consider the following classes of constraints:

System	Data	$P\%$	$R\%$	$F_1\%$	$E\%$
MANLI-Exact with M constraints	dev test	**86.8** **88.8**	84.5 85.7	**85.6** **87.2**	25.3 29.9
MANLI-Exact with L constraints	dev test	**86.1** **88.2**	84.6 86.4	**85.3** **87.3**	24.5 27.6
MANLI-Exact with M + L constraints	dev test	**87.1** **89.5**	84.4 86.2	**85.8** **87.8**	25.4 33.0

Table 3: Performance of MANLI-Exact featuring additional modifier (M) and lineage (L) constraints. Figures in boldface are statistically significant over the unconstrained MANLI reimplementation (p $\leq$ 0.05).

Modifier constraints: Tokens t that represent conjunctions, determiners, modals and cardinals can only be active if their parent tokens $\pi(t)$ are active.

$$\begin{aligned} & y_t - y_{\pi(t)} <= 0 \\ & \qquad \text{if POS}(t) \in \{\texttt{CC}, \texttt{CD}, \texttt{MD}, \texttt{DT}, \texttt{PDT}, \texttt{WDT}\} \end{aligned}$$

Lineage constraints: Tokens t that represent prepositions and particles (which are often confused by parsers) can only be active if one of their ancestors $\alpha(t)$ or descendants $\delta(t)$ is active. These constraints are less restrictive than the modifier constraints in order to account for attachment errors.

$$\begin{gathered} y_t - \sum_{a \in \alpha(t)} y_a - \sum_{d \in \delta(t)} y_d <= 0 \\ \text{if POS}(t) \in \{\texttt{IN}, \texttt{TO}, \texttt{RP}\} \end{gathered}$$

5.1 Alignment experiments

A TAG-based probabilistic dependency parser (Bangalore et al., 2009) is used to formulate the above constraints in our experiments. The results are shown in Table 3 and indicate a notable increase in alignment precision, which is to be expected as the constraints specifically seek to exclude poor edits. Despite the simple and overly general restrictions being applied, recall is almost unaffected. Most compellingly, the number of perfect alignments produced by the system increases significantly when

[2] Our Python reimplementation closely follows the original Java implementation of MANLI and was optimized for performance. MacCartney et al. (2008) report a decoding time of about 2 seconds per problem.

compared to the unconstrained models from Table 1 (a relative increase of 35% on the test corpus).

6 Discussion

The results of our evaluation indicate that exact decoding via ILP is a robust and efficient technique for solving alignment problems. Furthermore, the incorporation of simple constraints over a dependency parse can help to shape more accurate alignments. An examination of the alignments produced by our system reveals that many remaining errors can be tackled by the use of named-entity recognition and better paraphrase corpora; this was also noted by MacCartney et al. (2008) with regard to the original MANLI system. In addition, stricter constraints that enforce the alignment of syntactically-related tokens (rather than just their inclusion in the solution) may also yield performance gains.

Although MANLI's structured prediction approach to the alignment problem allows us to encode preferences as features and learn their weights via the structured perceptron, the decoding constraints used here can be used to establish dynamic links between alignment edits which cannot be determined *a priori*. The interaction between the selection of soft features for structured prediction and hard constraints for decoding is an interesting avenue for further research on this task. Initial experiments with a feature that considers the similarity of dependency heads of tokens in an edit (similar to MANLI's contextual features that look at preceding and following words) yielded some improvement over the baseline models; however, this did not perform as well as the simple constraints described above. Specific features that approximate soft variants of these constraints could also be devised but this was not explored here.

In addition to the NLI applications considered in this work, we have also employed the MANLI alignment technique to tackle alignment problems that are not inherently asymmetric such as the sentence fusion problems from McKeown et al. (2010). Although the absence of asymmetric alignment features affects performance marginally over the RTE2 dataset, all the performance gains exhibited by exact decoding with constraints appear to be preserved in symmetric settings.

7 Conclusion

We present a simple exact decoding technique as an alternative to approximate search-based decoding in MANLI that exhibits a twenty-fold improvement in runtime performance in our experiments. In addition, we propose novel syntactically-informed constraints to increase precision. Our final system improves over the results reported in MacCartney et al. (2008) by about 4.5% in precision and 1% in recall, with a large gain in the number of perfect alignments over the test corpus. Finally, we analyze the alignments produced and suggest that further improvements are possible through careful feature/constraint design, as well as the use of named-entity recognition and additional resources.

Acknowledgments

The authors are grateful to Bill MacCartney for providing a reference MANLI implementation and the anonymous reviewers for their useful feedback. This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-05-34871. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Srinivas Bangalore, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA: a probabilistic dependency parser based on tree insertion grammars. In *Proceedings of HLT-NAACL*, pages 185–188.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL Recognising Textual Entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Regina Barzilay and Lilian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of EMNLP*.

Chris Brockett. 2007. Aligning the 2006 RTE corpus. Technical Report MSR-TR-2007-77, Microsoft Research.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and

Christopher D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: an integer linear programming approach. *Journal of Artifical Intelligence Research*, 31:399–429, March.

Michael Collins. 2002. Discriminative training methods for hidden Markov models. In *Proceedings of EMNLP*, pages 1–8.

Hal Daumé, III and Daniel Marcu. 2005. Learning as search optimization: approximate large margin methods for structured prediction. In *Proceedings of ICML*, pages 169–176.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of ACL-HLT*, pages 25–28.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of EMNLP*, pages 177–185.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*, pages 104–111.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP*, pages 802–811.

André F. T. Martins, Noah A. Smith, and Eric P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*, pages 342–350.

Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Proceedings of HLT-NAACL*, pages 456–464.

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

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

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of EMNLP*, pages 142–149, July.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*, pages 836–841.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of EMNLP*, pages 513–523.