A Phrase-Based Alignment Model for Natural Language Inference

Bill MacCartney, Michel Galley, Christopher D. Manning
Natural Language Processing Group, Stanford University
{wcmac,mgalley,manning}@stanford.edu

Abstract

The alignment problem—establishing links between corresponding phrases in two related sentences—is as important in natural language inference (NLI) as it is in machine translation (MT). But the tools and techniques of MT alignment do not readily transfer to NLI, where one cannot assume semantic equivalence, and for which large volumes of bitext are lacking. We present a new NLI aligner, the MANLI system, designed to address these challenges. It uses a phrase-based alignment representation, exploits external lexical resources, and capitalizes on a new set of supervised training data. We compare the performance of MANLI to existing NLI and MT aligners on an NLI alignment task over the well-known Recognizing Textual Entailment data. We show that MANLI significantly outperforms existing aligners, achieving gains of 6.2% in F_1 over a representative NLI aligner and 10.5% over GIZA++.

1 Introduction

The problem of natural language inference (NLI) is to determine whether a natural-language hypothesis H can reasonably be inferred from a given premise text P. In order to recognize that *Kennedy was killed* can be inferred from *JFK was assassinated*, one must first recognize the correspondence between *Kennedy* and *JFK*, and between *killed* and *assassinated*. Consequently, most current approaches to NLI rely, implicitly or explicitly, on a facility for *alignment*—that is, establishing links between corresponding entities and predicates in P and H. Recent entries in the annual Recognizing Textual Entailment (RTE) competition (Dagan et al., 2005) have addressed the alignment problem in a variety of ways, though often without distinguishing it as a separate subproblem. Glickman et al. (2005) and Jijkoun and de Rijke (2005), among others, have explored approaches based on measuring the degree of lexical overlap between bags of words. While ignoring structure, such methods depend on matching each word in H to the word in P with which it is most similar—in effect, an alignment. At the other extreme, Tatu and Moldovan (2007) and Bar-Haim et al. (2007) have formulated the inference problem as analogous to proof search, using inferential rules which encode (among other things) knowledge of lexical relatedness. In such approaches, the correspondence between the words of P and H is implicit in the steps of the proof.

Increasingly, however, the most successful RTE systems have made the alignment problem explicit. Marsi and Krahmer (2005) and MacCartney et al. (2006) first advocated pipelined system architectures containing a distinct alignment component, a strategy crucial to the top-performing systems of Hickl et al. (2006) and Hickl and Bensley (2007). However, each of these systems has pursued alignment in idiosyncratic and poorly-documented ways, often using proprietary data, making comparisons and further development difficult.

In this paper we undertake the first systematic study of alignment for NLI. We propose a new NLI alignment system which uses a phrase-based representation of alignment, exploits external resources for knowledge of semantic relatedness, and capitalizes on the recent appearance of new supervised training data for NLI alignment. In addition, we examine the relation between NLI alignment and MT alignment, and investigate whether existing MT aligners can usefully be applied in the NLI setting.

2 NLI alignment vs. MT alignment

The alignment problem is familiar in machine translation (MT), where recognizing that *she came* is a good translation for *elle est venue* requires establish-

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 802–811,
Honolulu, October 2008.

ing a correspondence between *she* and *elle*, and between *came* and *est venue*. The MT community has developed not only an extensive literature on alignment (Brown et al., 1993; Vogel et al., 1996; Marcu and Wong, 2002; DeNero et al., 2006), but also standard, proven alignment tools such as GIZA++ (Och and Ney, 2003). Can off-the-shelf MT aligners be applied to NLI? There is reason to be doubtful. Alignment for NLI differs from alignment for MT in several important respects, including:

1. Most obviously, it is monolingual rather than cross-lingual, opening the door to utilizing abundant (monolingual) sources of information on semantic relatedness, such as WordNet.
2. It is intrinsically asymmetric: P is often much longer than H, and commonly contains phrases or clauses which have no counterpart in H.
3. Indeed, one cannot assume even approximate semantic equivalence—usually a given in MT. Because NLI problems include both valid and invalid inferences, the semantic content of H may diverge substantially from P. An NLI aligner must be designed to accommodate frequent unaligned words and phrases.
4. Little training data is available. MT alignment models are typically trained in unsupervised fashion, inducing lexical correspondences from massive quantities of sentence-aligned bitexts. While NLI aligners could in principle do the same, large volumes of suitable data are lacking. NLI aligners must therefore depend on smaller quantities of supervised training data, supplemented by external lexical resources. Conversely, while existing MT aligners can make use of dictionaries, they are not designed to harness other sources of information on degrees of semantic relatedness.

Consequently, the tools and techniques of MT alignment may not transfer readily to NLI alignment. We investigate the matter empirically in section 5.2.

3 Data

Until recently, research on alignment for NLI has been hampered by a paucity of high-quality, publicly available data from which to learn. Happily, that has begun to change, with the release by Microsoft Research (MSR) of human-generated alignment annotations (Brockett, 2007) for inference problems from the second Recognizing Textual Entailment (RTE2) challenge (Bar-Haim et al., 2006). To our knowledge, this work is the first to exploit this data for training and evaluation of NLI alignment models.

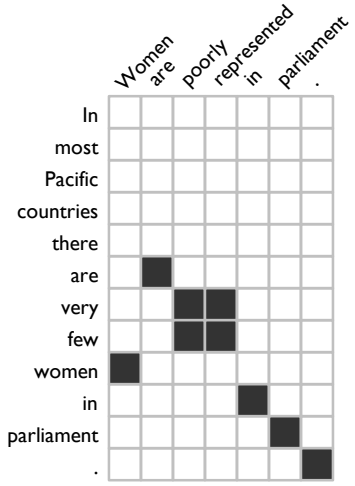

Figure 1: The MSR gold-standard alignment for problem 116 from the RTE2 development set.

The RTE2 data consists of a development set and a test set, each containing 800 inference problems. Each problem consists of a premise and a hypothesis. The premises contain 29 words on average; the hypotheses, 11 words. Each problem is marked as a valid or invalid inference (50% each); however, these annotations are ignored during alignment, since they would not be available during testing of a complete NLI system.

The MSR annotations use an alignment representation which is token-based, but many-to-many, and thus allows implicit alignment of multi-word phrases. Figure 1 shows an example in which *very few* has been aligned with *poorly represented*.

In the MSR data, every alignment link is marked as SURE or POSSIBLE. In making this distinction, the annotators have followed a convention common in MT, which permits alignment precision to be measured against both SURE and POSSIBLE links, while recall is measured against only SURE links. In this work, however, we have chosen to ignore POSSIBLE links, embracing the argument made by (Fraser and Marcu, 2007) that their use has impeded progress in MT alignment models, and that SURE-

only annotation is to be preferred.

Each RTE2 problem was independently annotated by three people, following carefully designed annotation guidelines. Inter-annotator agreement was high: Brockett (2007) reports Fleiss' kappa[1] scores of about 0.73 ("substantial agreement") for mappings from H tokens to P tokens; and all three annotators agreed on $\sim$70% of proposed links, while at least two of three agreed on more than 99.7% of proposed links,[2] attesting to the high quality of the annotation data. For this work, we merged the three independent annotations, using majority rule,[3] to obtain a gold-standard annotation containing an average of 7.3 links per RTE problem.

4 The MANLI aligner

In this section, we describe the MANLI aligner, a new alignment system designed expressly for NLI alignment. The MANLI system consists of four elements: (1) a phrase-based representation of alignment, (2) a feature-based linear scoring function for alignments, (3) a decoder which uses simulated annealing to find high-scoring alignments, and (4) perceptron learning to optimize feature weights.

4.1 A phrase-based alignment representation

MANLI uses an alignment representation which is intrinsically phrase-based. (Following the usage common in MT, we use "phrase" to mean any contiguous span of tokens, not necessarily corresponding to a syntactic phrase.) We represent an alignment E between a premise P and a hypothesis H as a set of phrase edits $\{e_1, e_2, \ldots\}$, each belonging to one of four types:

- an EQ edit connects a phrase in P with an equal (by word lemmas) phrase in H
- a SUB edit connects a phrase in P with an unequal phrase in H
- a DEL edit covers an unaligned phrase in P
- an INS edit covers an unaligned phrase in H

For example, the alignment shown in figure 1 can be represented by the set $\{\text{DEL}(\textit{In}_1)$, $\text{DEL}(\textit{most}_2)$, $\text{DEL}(\textit{Pacific}_3)$, $\text{DEL}(\textit{countries}_4)$, $\text{DEL}(\textit{there}_5)$, $\text{EQ}(\textit{are}_6, \textit{are}_2)$, $\text{SUB}(\textit{very}_7\ \textit{few}_8, \textit{poorly}_3\ \textit{represented}_4)$, $\text{EQ}(\textit{women}_9, \textit{Women}_1)$, $\text{EQ}(\textit{in}_{10}, \textit{in}_5)$, $\text{EQ}(\textit{parliament}_{11}, \textit{parliament}_6)$, $\text{EQ}(._{12}, ._7)\}$.[4]

Alignments are constrained to be one-to-one at the phrase level: every token in P and H belongs to exactly one phrase, which participates in exactly one edit (possibly DEL or INS). However, the phrase representation permits alignments which are many-to-many at the token level. In fact, this is the chief motivation for the phrase-based representation: we can align *very few* and *poorly represented* as units, without being forced to make an arbitrary choice as to which word goes with which word. Moreover, our scoring function can make use of lexical resources which have information about semantic relatedness of multi-word phrases, not merely individual words.

About 23% of the MSR gold-standard alignments are not one-to-one (at the token level), and are therefore technically unreachable for MANLI, which is constrained to generate one-to-one alignments. However, by merging contiguous token links into phrase edits of size > 1, most MSR alignments (about 92%) can be straightforwardly converted into MANLI-reachable alignments. For the purpose of model training (but *not* for the evaluation described in section 5.4), we generated a version of the MSR data in which all alignments were converted to MANLI-reachable form.[5]

4.2 A feature-based scoring function

To score alignments, we use a simple feature-based linear scoring function, in which the score of an alignment is the sum of the scores of the edits it contains (including not only SUB and EQ edits, but also DEL and INS edits), and the score of an edit is the dot product of a vector encoding its features and a vector of weights. If E is a set of edits constituting

[1] Fleiss' kappa generalizes Cohen's kappa to the case where there are more than two annotators.

[2] The SURE/POSSIBLE distinction is taken as significant in computing all these figures.

[3] The handful of three-way disagreements were treated as POSSIBLE links, and thus were not used here.

[4] DEL and INS edits of size > 1 are possible in principle, but are not used in our training data.

[5] About 8% of the MSR alignments contain non-contiguous links, most commonly because P contains two references to an entity (e.g., *Christian Democrats* and *CDU*) which are both linked to a reference to the same entity in H (e.g., *Christian Democratic Union*). In such cases, one or more links must be eliminated to achieve a MANLI-reachable alignment. We used a string-similarity heuristic to break such conflicts, but were obliged to make an arbitrary choice in about 2% of cases.

an alignment, and $\mathbf{\Phi}$ is a vector of feature functions, the score s is given by:

$$s(E) = \sum_{e \in E} s(e) = \sum_{e \in E} \mathbf{w} \cdot \mathbf{\Phi}(e)$$

We'll explain how the feature weights $\mathbf{w}$ are set in section 4.4. The features used to characterize each edit are as follows:

Edit type features. We begin with boolean features encoding the type of each edit. We expect EQs to score higher than SUBs, and (since P is commonly longer than H) DELs to score higher than INSs.

Phrase features. Next, we have features which encode the sizes of the phrases involved in the edit, and whether these phrases are non-constituents (in syntactic parses of the sentences involved).

Lexical similarity feature. For SUB edits, a very important feature represents the lexical similarity of the substituends, as a real value in $[0, 1]$. This similarity score is computed as a max over a number of component scoring functions, some based on external lexical resources, including:

- various string similarity functions, of which most are applied to word lemmas
- measures of synonymy, hypernymy, antonymy, and semantic relatedness, including a widely-used measure due to Jiang and Conrath (1997), based on manually constructed lexical resources such as WordNet and NomBank
- a function based on the well-known distributional similarity metric of Lin (1998), which automatically infers similarity of words and phrases from their distributions in a very large corpus of English text

The ability to leverage external lexical resources—both manually and automatically constructed—is critical to the success of MANLI.

Contextual features. Even when the lexical similarity for a SUB edit is high, it may not be a good match. If P or H contains multiple occurrences of the same word—which happens frequently with function words, and occasionally with content words—lexical similarity may not suffice to determine the right match. To remedy this, we introduce contextual features for SUB and EQ edits. A real-valued *distortion* feature measures the difference between the relative positions of the substituends within their respective sentences, while boolean *matching neighbors* features indicate whether the tokens before and after the substituends are equal or similar.

Inputs

- an alignment problem $\langle P, H \rangle$
- a number of iterations N (e.g. 100)
- initial temperature T_0 (e.g. 40) and multiplier r (e.g. 0.9)
- a bound on edit size max (e.g. 6)
- an alignment scoring function, SCORE(E)

Initialize

- Let E be an "empty" alignment for $\langle P, H \rangle$ (containing only DEL and INS edits, no EQ or SUB edits)
- Set $\hat{E} = E$

Repeat for i = 1 to N

- Let $\{F_1, F_2, ...\}$ be the set of possible successors of E. To generate this set:
  - Consider every possible edit f up to size max
  - Let $C(E, f)$ be the set of edits in E which "conflict" with f (i.e., involve at least some of the same tokens as f)
  - Let $F = E \cup \{f\} \setminus C(E, f)$
- Let $s(F)$ be a map from successors of E to scores generated by SCORE
- Set $p(F) = \exp s(F)$, and then normalize $p(F)$, transforming the score map to a probability distribution
- Set $T_i = r \cdot T_{i-1}$
- Set $p(F) = p(F)^{1/T_i}$, smoothing or sharpening $p(F)$
- Renormalize $p(F)$
- Choose a new value for E by sampling from $p(F)$
- If SCORE(E) > SCORE($\hat{E}$), set $\hat{E}$ = E

Return $\hat{E}$

Figure 2: The MANLI-ALIGN algorithm

4.3 Decoding using simulated annealing

The problem of decoding—that is, finding a high-scoring alignment for a particular inference problem—is made more complex by our choice of a phrase-based alignment representation. For a model which uses a token-based representation (say, one which simply maps H tokens to P tokens), decoding is trivial, since each token can be aligned independently of its neighbors. (This is the case for the bag-of-words aligner described in section 5.1.) But with a phrase-based representation, things are more complicated. The segmentation into phrases is not given in advance, and every phrase pair considered for alignment must be consistent with its neighbors with respect to segmentation. Consequently, the decoding problem cannot be factored into a number of

independent decisions.

To address this difficulty, we have devised a stochastic alignment algorithm, MANLI-ALIGN (figure 2), which uses a simulated annealing strategy. Beginning from an arbitrary alignment, we make a series of local steps, at each iteration sampling from a set of possible successors according to scores assigned by our scoring function. The sampling is controlled by a "temperature" which falls over time. At the beginning of the process, successors are sampled with nearly uniform probability, which helps to ensure that the space of possibilities is explored and local maxima are avoided. As the temperature falls, there is a ever-stronger bias toward high-scoring successors, so that the algorithm converges on a near-optimal alignment. Clever use of memoization helps to ensure that computational costs remain manageable. Using the parameter values suggested in figure 2, aligning an average RTE problem takes about two seconds.

While MANLI-ALIGN is not guaranteed to produce optimal alignments, there is reason to believe that it usually comes very close. After training, the alignment found by MANLI scored at least as high as the gold alignment for 99.6% of RTE problems.[6]

4.4 Perceptron learning

To tune the parameters **w** of the model, we use an adaptation of the averaged perceptron algorithm (Collins, 2002), which has proven successful on a range of NLP tasks. The algorithm is shown in figure 3. After initializing **w** to 0, we perform N training epochs. (Our experiments used $N = 50$.) In each epoch, we iterate through the training data, updating the weight vector at each training example according to the difference between the features of the target alignment and the features of the alignment produced by the decoder using the current weight vector. The size of the update is controlled by a learning rate which decreases over time. At the end of each epoch, the weight vector is normalized and stored. The final result is the average of the stored weight vectors, omitting vectors from a fixed number of epochs at the beginning of the run (which tend to be of poor quality). Using the parameter values suggested in figure 3, training runs on the RTE2 development set required about 20 hours.

Inputs

- training problems $\langle P_j, H_j \rangle$, $j = 1..n$
- corresponding gold-standard alignments E_j
- a number of learning epochs N (e.g. 50)
- a "burn-in" period $N_0 < N$ (e.g. 10)
- initial learning rate R_0 (e.g. 1) and multiplier r (e.g. 0.8)
- a vector of feature functions $\mathbf{\Phi}(E)$
- an alignment algorithm ALIGN$(P, H; \mathbf{w})$ which finds a good alignment for $\langle P, H \rangle$ using weight vector **w**

Initialize

- Set $\mathbf{w} = 0$

Repeat for i = 1 to N

- Set $R_i = r \cdot R_{i-1}$, reducing the learning rate
- Randomly shuffle the training problems
- For j = 1 to n:
  - Set $\hat{E}_j$ = ALIGN$(P_j, H_j; \mathbf{w})$
  - Set $\mathbf{w} = \mathbf{w} + R_i \cdot (\mathbf{\Phi}(E_j) - \mathbf{\Phi}(\hat{E}_j))$
- Set $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|_2$ (L2 normalization)
- Set $\mathbf{w}[i] = \mathbf{w}$, storing the weight vector for this epoch

Return an averaged weight vector:

- $\mathbf{w}_{avg} = 1/(N - N_0) \sum_{i=N_0+1}^{N} \mathbf{w}[i]$

Figure 3: The MANLI-LEARN algorithm

5 Evaluating aligners on MSR data

In this section, we describe experiments designed to evaluate the performance of various alignment systems on the MSR gold-standard data described in section 3. For each system, we report precision, recall, and F-measure (F_1).[7] Note that these are macro-averaged statistics, computed per problem by counting aligned token pairs,[8] and then averaged over all problems in a problem set.[9] We also re-

[6] This figure is based on the MANLI-reachable version of the gold-standard data described in section 4.1. For the raw gold-standard data, the figure is 88.1%. The difference is almost entirely attributable to unreachable gold alignments, which tend to score higher simply because they contain more edits (and because the learned weights are mostly positive).

[7] MT researchers conventionally report results in terms of alignment error rate (AER). Since we use only SURE links in the gold-standard data (see section 3), AER is equivalent to $1 - F_1$.

[8] For phrase-based alignments like those generated by MANLI, two tokens are considered to be aligned iff they are contained within phrases which are aligned.

[9] MT evaluations conventionally use micro-averaging, which gives greater weight to problems containing more aligned pairs. This makes sense in MT, where the purpose of alignment is to induce phrase tables. But in NLI, where the ultimate goal is to maximize the number of inference problems answered correctly, it is more fitting to give all problems equal weight, and so we macro-average. We have also generated all results using micro-averaging, and found that the relative comparisons are

port the exact match rate, that is, the proportion of problems in which the guessed alignment exactly matches the gold alignment. The results are summarized in table 1.

5.1 A robust baseline: the bag-of-words aligner

As a baseline, we use a simple alignment algorithm inspired by the lexical entailment model of Glickman et al. (2005), and similar to the simple heuristic model described in (Och and Ney, 2003). Each hypothesis word h is aligned to the premise word p to which it is most similar, according to a lexical similarity function $sim(p, h)$ which returns scores in $[0, 1]$. While Glickman et al. used a function based on web co-occurrence statistics, we use a much simpler function based on string edit distance:

$$sim(w_1, w_2) = 1 - \frac{dist(lem(w_1), lem(w_2))}{max(|lem(w_1)|, |lem(w_2)|)}$$

(Here $lem(w)$ denotes the lemma of word w; $dist()$ denotes Levenshtein string edit distance; and $|\cdot|$ denotes string length.)

This model can be easily extended to generate an alignment score, which will be of interest in section 6. We define the score for a specific hypothesis token h to be the log of its similarity with the premise token p to which it is aligned, and the score for the complete alignment of hypothesis H to premise P to be the sum of the scores of the tokens in H, weighted by *inverse document frequency* in a large corpus[10] (so that common words get less weight), and normalized by the length of H:

$$score(h|P) = \log \max_{p \in P} sim(p, h)$$

$$score(H|P) = \frac{1}{|H|} \sum_{h \in H} idf(h) \cdot score(h|P)$$

Despite the simplicity of this alignment model, its performance is fairly robust, with good recall. Its precision, however, its mediocre—chiefly because, by design, it aligns every h with some p. The model could surely be improved by allowing it to leave some H tokens unaligned, but this was not pursued.

not greatly affected.

[10] We use $idf(w) = \log(N/N_w)$, where N is the number of documents in the corpus, and N_w is the number of documents containing word w.

System	Data	P %	R %	F_1 %	E %
Bag-of-words	dev	57.8	81.2	67.5	3.5
(baseline)	test	62.1	82.6	70.9	5.3
GIZA++	dev	83.0	66.4	72.1	9.4
(using lex, $\cap$)	test	85.1	69.1	74.8	11.3
Cross-EM	dev	67.6	80.1	72.1	1.3
(using lex, $\cap$)	test	70.3	81.0	74.1	0.8
Stanford RTE	dev	81.1	61.2	69.7	0.5
	test	82.7	61.2	70.3	0.3
Stanford RTE	dev	81.1	75.8	78.4	—
(punct. corr.)	test	82.7	75.8	79.1	—
MANLI	dev	83.4	85.5	84.4	21.7
(this work)	test	85.4	85.3	85.3	21.3

Table 1: Performance of various aligners on the MSR RTE2 alignment data. The columns show the data set used (800 problems each); average precision, recall, and F-measure; and the exact match rate (see text).

5.2 MT aligners: GIZA++ and Cross-EM

Given the importance of alignment for NLI, and the availability of standard, proven tools for MT alignment, an obvious question presents itself: why not use an off-the-shelf MT aligner for NLI? Although we have argued (section 2) that this is unlikely to succeed, to our knowledge, we are the first to investigate the matter empirically.[11]

The best-known MT aligner is undoubtedly GIZA++ (Och and Ney, 2003), which contains implementations of various IBM models (Brown et al., 1993), as well as the HMM model of Vogel et al. (1996). Most practitioners use GIZA++ as a black box, via the Moses MT toolkit (Koehn et al., 2007). We followed this practice, running with Moses' default parameters on the RTE2 data to obtain asymmetric word alignments in both directions (P-to-H and H-to-P). We then performed symmetrization using the well-known INTERSECTION heuristic.

Unsurprisingly, the out-of-the-box performance was quite poor, with most words aligned apparently at random. Precision was fair (72%) but recall was very poor (46%). Even equal words were usually not aligned—because GIZA++ is designed for cross-linguistic use, it does not consider word equality between source and target sentences. To remedy this, we supplied GIZA++ with a lexicon, using a trick

[11] However, Dolan et al. (2004) explore a closely-related topic: using an MT aligner to identify paraphrases.

common in MT: we supplemented the training data with synthetic data consisting of matched pairs of equal words. This gives GIZA++ a better chance of learning that, e.g., *man* should align with *man*. The result was a big boost in recall (+23%), and a smaller gain in precision. The results for GIZA++ shown in table 1 are based on using the lexicon and INTERSECTION. With these settings, GIZA++ properly aligned most pairs of equal words, but continued to align other words apparently at random.

Next, we compared the performance of INTERSECTION with other symmetrization heuristics defined in Moses—including UNION, GROW, GROW-DIAG, GROW-DIAG-FINAL (the default), and GROW-DIAG-FINAL-AND—and with asymmetric alignments in both directions. While all these alternatives achieved better recall than INTERSECTION, all showed substantially worse precision and F_1. On the RTE2 test set, the asymmetric alignment from H to P scored 68% in F_1; GROW scored 58%; and all other alternatives scored below 52%.

As an additional experiment, we tested the Cross-EM aligner (Liang et al., 2006) from the BerkeleyAligner package on the MSR data. While this aligner is in many ways simpler than GIZA++ (it lacks any model of fertility, for example), its method of jointly training two simple asymmetric HMM models has outperformed GIZA++ on standard evaluations of MT alignment. As with GIZA++, we experimented with a variety of symmetrization heuristics, and ran trials with and without a supplemental lexicon. The results were broadly similar: INTERSECTION greatly outperformed alternative heuristics, and using a lexicon provided a big boost (up to 12% in F_1). Under optimal settings, the Cross-EM aligner showed better recall and worse precision than GIZA++, with F_1 just slightly lower. Like GIZA++, it did well at aligning equal words, but aligned most other words at random.

The mediocre performance of MT aligners on NLI alignment comes as no surprise, for reasons discussed in section 2. Above all, the quantity of training data is simply too small for unsupervised learning to succeed. A successful NLI aligner will need to exploit supervised training data, and will need access to additional sources of knowledge about lexical relatedness.

5.3 The Stanford RTE aligner

A better comparison is thus to an alignment system expressly designed for NLI. For this purpose, we used the alignment component of the Stanford RTE system (Chambers et al., 2007). The Stanford aligner performs decoding and learning in a similar fashion to MANLI, but uses a simpler, token-based alignment representation, along with a richer set of features for alignment scoring. It represents alignments as an injective map from H tokens to P tokens. Phrase alignments are not directly representable, although the effect can be approximated by a pre-processing step which collapses multi-token named entities and certain collocations into single tokens. The features used for alignment scoring include not only measures of lexical similarity, but also syntactic features intended to promote the alignment of similar predicate-argument structures.

Despite this sophistication, the out-of-the-box performance of the Stanford aligner is mediocre, as shown in table 1. The low recall figures are particularly noteworthy. However, a partial explanation is readily available: by design, the Stanford system ignores punctuation.[12] Because punctuation tokens constitute about 15% of the aligned pairs in the MSR data, this sharply reduces measured recall. However, since punctuation matters little in inference, such recall errors probably should be forgiven. Thus, table 1 also shows adjusted statistics for the Stanford system in which all recall errors involving punctuation are (generously) ignored.

Even after this adjustment, the recall figures are unimpressive. Error analysis reveals that the Stanford aligner does a poor job of aligning function words. About 13% of the aligned pairs in the MSR data are matching prepositions or articles; the Stanford aligner misses about 67% of such pairs. (By contrast, MANLI misses only 10% of such pairs.) While function words matter less in inference than nouns and verbs, they are not irrelevant, and because sentences often contain multiple instances of a particular function word, matching them properly is by no means trivial. If matching prepositions and articles were ignored (in addition to punctuation), the gap in F_1 between the MANLI and Stanford systems

[12] In fact, it operates on a dependency-graph representation from which punctuation is omitted.

would narrow to about 2.8%.

Finally, the Stanford aligner is handicapped by its token-based alignment representation, often failing (partly or completely) to align multi-word phrases such as *peace activists* with *protesters*, or *hackers* with *non-authorized personnel*.

5.4 The MANLI aligner

As table 1 indicates, the MANLI aligner was found to outperform all other aligners evaluated on every measure of performance, achieving an F_1 score 10.5% higher than GIZA++ and 6.2% higher than the Stanford aligner (even with the punctuation correction).[13] MANLI achieved a good balance between precision and recall, and matched more than 20% of the gold-standard alignments exactly.

Three factors seem to have contributed most to MANLI's success. First, MANLI is able to outperform the MT aligners principally because it is able to leverage lexical resources to identify the similarity between pairs of words such as *jail* and *prison*, *prevent* and *stop*, or *injured* and *wounded*. Second, MANLI's contextual features enable it to do better than the Stanford aligner at matching function words, a weakness of the Stanford aligner discussed in section 5.3. Third, MANLI gains a marginal advantage because its phrase-based representation of alignment permits it to properly align phrase pairs such as *death penalty* and *capital punishment*, or *abdicate* and *give up*.

However, the phrase-based representation contributed far less than we had hoped. Setting MANLI's maximum phrase size to 1 (effectively, restricting it to token-based alignments) caused F_1 to fall by just 0.2%. We do not interpret this to mean that phrase alignments are not useful—indeed, about 2.6% of the links in the gold-standard data involve phrases of size > 1. Rather, we think it shows that we have failed to fully exploit the advantages of the phrase-based representation, chiefly because we lack lexical resources providing good information on similarity of multi-word phrases.

Error analysis suggests that there is ample room for improvement. A large proportion of recall errors (perhaps 40%) occur because the lexical similarity function assigns too low a value to pairs of words or phrases which are clearly similar, such as *conservation* and *protecting*, *server* and *computer networks*, *organization* and *agencies*, or *bone fragility* and *osteoporosis*. Better exploitation of lexical resources could help to reduce such errors. Another important category of recall errors (about 12%) result from the failure to identify one- and multi-word versions of the name of some entity, such as *Lennon* and *John Lennon*, or *Nike Inc.* and *Nike*. A special-purpose similarity function could help here. Note, however, that about 10% of recall errors are unavoidable, given our choice of alignment representation, since they involve cases where the gold standard aligns one or more tokens on one side to a non-contiguous set of tokens on the other side.

Precision errors may be harder to reduce. These errors are dominated by cases where we mistakenly align two equal function words (49% of precision errors), two forms of the verb *to be* (21%), two equal punctuation marks (7%), or two words or phrases of other types having equal lemmas (18%). Because such errors often occur because the aligner is forced to choose between nearly equivalent alternatives, they may be difficult to eliminate. The remaining 5% of precision errors result mostly from aligning words or phrases rightly judged to be highly similar, such as *expanding* and *increasing*, *labor* and *birth*, *figures* and *number*, or *223,000* and *220,000*.

6 Using alignment to predict RTE answers

In section 5, we evaluated the ability of aligners to recover gold-standard alignments. But since alignment is just one component of the NLI problem, we might also examine the impact of different aligners on the ability to recognize valid inferences. If a high-scoring alignment indicates a close correspondence between H and P, does this also indicate a valid inference? We have previously emphasized (MacCartney et al., 2006) that there is more to inferential validity than close lexical or structural correspondence: negations, modals, non-factive and implicative verbs, and other linguistic constructs can affect validity in ways hard to capture in alignment. Nevertheless, alignment score can be a strong predictor of inferential validity, and some NLI systems (e.g., (Glickman et al., 2005)) rely entirely on some measure of alignment quality to predict validity.

[13] Reported results for MANLI are averages over 10 runs.

System	data	acc %	avgP %
Bag-of-words aligner	dev	61.3	61.5
	test	57.9	58.9
Stanford RTE aligner	dev	63.1	64.9
	test	60.9	59.2
MANLI aligner	dev	59.3	69.0
(this work)	test	60.3	61.0
RTE2 entries (average)	test	58.5	59.1
LCC (Hickl et al., 2006)	test	75.4	80.8

Table 2: Performance of various aligners and complete RTE systems in predicting RTE2 answers. The columns show the data set used, accuracy, and average precision (the recommended metric for RTE2).

If an aligner generates real-valued alignment scores, we can use the RTE data to test its ability to predict inferential validity with the following simple method. For a given RTE problem, we predict YES (valid) if its alignment score[14] exceeds a threshold τ, and NO otherwise. We tune τ to maximize accuracy on the RTE2 development set, and then measure performance on the RTE2 test set using the same τ.

Table 2 shows results for several NLI aligners, along with some results for complete RTE systems, including the LCC system (the top performer at RTE2) and an average of all systems participating in RTE2. While none of the aligners rivals the performance of the LCC system, all achieve respectable results, and the Stanford and MANLI aligners outperform the average RTE2 entry. Thus, even if alignment quality does not determine inferential validity, many NLI systems could be improved by harnessing a well-designed NLI aligner.

7 Related work

Given the extensive literature on phrase-based MT, it may be helpful further to situate our phrase-based alignment model in relation to past work. The standard approach to training a phrase-based MT system is to apply phrase extraction heuristics using word-aligned training sets (Och and Ney, 2003; Koehn et al., 2007). Unfortunately, word alignment models assume that source words are individually translated into target words, which stands at odds with the key assumption in phrase-based systems that many translations are non-compositional. More recently, several works (Marcu and Wong, 2002; DeNero et al., 2006; Birch et al., 2006; DeNero and Klein, 2008) have presented more unified phrase-based systems that jointly align and weight phrases, though these systems have not come close to the state of the art when evaluated in terms of MT performance.

We would argue that previous work in MT phrase alignment is orthogonal to our work. In MANLI, the need for phrases arises when word-based representations are not appropriate for alignment (e.g., between *close down* and *terminate*), though longer phrases are not needed to achieve good alignment quality. In MT phrase alignment, it is beneficial to account for arbitrarily large phrases, since the larger contexts offered by these phrases can help realize more dependencies among translated words (e.g., word order, agreement, subcategorization). Perhaps because MT phrase alignment is dealing with much larger contexts, no existing work in MT phrase alignment (to our knowledge) directly models word insertions and deletions, as in MANLI. For example, in figure 1, MANLI can just skip *In most Pacific countries there*, while an MT phrase-based model would presumably align *In most Pacific countries there are* to *Women are*. Hence, previous work is of limited applicability to our problem.

8 Conclusion

While MT aligners succeed by unsupervised learning of word correspondences from massive amounts of bitext, NLI aligners are forced to rely on smaller quantities of supervised training data. With the MANLI system, we have demonstrated how to overcome this lack of data by utilizing external lexical resources, and how to gain additional power from a phrase-based representation of alignment.

Acknowledgements The authors wish to thank the anonymous reviewers for their helpful comments on an earlier draft of this paper. This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM and in part by the CIA ATP as part of the OCCAM project.

[14]For good results, it may be necessary to normalize the alignment score. Scores from MANLI were normalized by the number of tokens in the problem. The Stanford aligner performs a similar normalization internally.

References

R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

R. Bar-Haim, I. Dagan, I. Greental, and E. Shnarch. 2007. Semantic Inference at the Lexical-Syntactic Level. In *Proceedings of AAAI-07*.

A. Birch, C. Callison-Burch, M. Osborne, and P. Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. In *Proceedings of the ACL-06 Workshop on Statistical Machine Translation*.

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

P. F. Brown, S. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

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-07 Workshop on Textual Entailment and Paraphrasing*.

M. Collins. 2002. Discriminative training methods for hidden Markov models. In *Proceedings of EMNLP-02*.

I. Dagan, O. Glickman, and B. Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

J. DeNero and D. Klein. 2008. The Complexity of Phrase Alignment Problems. In *Proceedings of ACL/HLT-08: Short Papers*, pages 25–28.

J. DeNero, D. Gillick, J. Zhang, and D. Klein. 2006. Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings of the ACL-06 Workshop on Statistical Machine Translation*, pages 31–38.

B. Dolan, C. Quirk, and C. Brockett. 2004. Unsupervised construction of large paraphrase corpora. In *Proceedings of COLING-04*.

A. Fraser and D. Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3):293–303.

O. Glickman, I. Dagan, and M. Koppel. 2005. Web based probabilistic textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*.

A. Hickl and J. Bensley. 2007. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *ACL-07 Workshop on Textual Entailment and Paraphrasing*, Prague.

A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi. 2006. Recognizing Textual Entailment with LCC's GROUNDHOG System. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*.

V. Jijkoun and M. de Rijke. 2005. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 73–76.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL-07, demonstration session*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by Agreement. In *Proceedings of NAACL-06*, New York.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*, pages 768–774, Montreal, Canada.

B. MacCartney, T. Grenager, M. C. de Marneffe, D. Cer, and C. D. Manning. 2006. Learning to Recognize Features of Valid Textual Entailments. In *Proceedings of NAACL-06*, New York.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP-02*, pages 133–139.

E. Marsi and E. Krahmer. 2005. Classification of semantic relations by humans and machines. In *ACL-05 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor.

F. J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

M. Tatu and D. Moldovan. 2007. COGEX at RTE3. In *Proceedings of ACL-07*.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING-96*, pages 836–841, Copenhagen, Denmark.