Relation Alignment for Textual Entailment Recognition

Mark Sammons, V.G.Vinod Vydiswaran, Tim Vieira, Nikhil Johri,
Ming-Wei Chang, Dan Goldwasser, Vivek Srikumar, Gourab Kundu,
Yuancheng Tu, Kevin Small, Joshua Rule, Quang Do, Dan Roth
Department of Computer Science
University of Illinois at Urbana-Champaign
{mssammon|vgvinodv|danr}@illinois.edu

Abstract

We present an approach to textual entailment recognition, in which inference is based on a shallow semantic representation of relations (predicates and their arguments) in the text and hypothesis of the entailment pair, and in which specialized knowledge is encapsulated in modular components with very simple interfaces. We propose an architecture designed to integrate different, unscaled Natural Language Processing resources, and demonstrate an alignment-based method for combining them. We clarify the purpose of alignment in the RTE task, identifying two distinct alignment models, each of which leads to a different type of entailment system. We identify desirable properties of alignment, and use this to inform our implementation of an alignment component. We evaluate the resulting system on the RTE5 data set, and use an ablation study to assess the conformance of our alignment approach with these desired characteristics.

1 Introduction

Machine Learning solutions for the problem of Recognizing Textual Entailment (RTE) must overcome a significant obstacle: the relatively small amount of labeled entailment data, given the complexity of the RTE problem. Previous successful approaches typically overcome this by using a pipeline model that extracts a limited number of high-level features from induced representations of entailment pairs, and training a classifier using labeled entailment corpora. In particular, these systems define some kind of alignment between the text and hypothesis of an entailment pair, and use this to simplify the machine learning problem. However, their treatments of alignment are in some cases not well specified, and in others do not account well for the need to incorporate a wider range of analytical resources in a straightforward manner. Moreover, the relationship between alignment and entailment is still unclear.

We attempt to address these and other problems using a meaning representation centered around natural language relations (predicate-argument structures that use natural language rather than canonical logical symbols), and an alignment-based model that integrates them with analytic resources operating at a range of granularities. We clarify possible roles for alignment in RTE systems, outline desirable characteristics for alignment, and implement a solution accordingly. We evaluate our implementation on the RTE5 data set, and show via an ablation study that our alignment component has some of the desirable characteristics we identify.

2 Alignment and Entailment

As observed in (MacCartney et al., 2008), alignment components are present in many of the more successful RTE systems. Some alignment-like mechanism seems essential for systems with supervised machine learning components: if local decisions are to play a role in a global decision, systems need signals either to identify the correct local comparisons to make, or to signify what label these local decisions should take when training the system. RTE data offers only a single global label, and alignment is a natural way to try to propagate this information to local decisions.

In order to facilitate progress on the general problem of alignment as it relates to Textual Entailment, we examine ways in which alignment has been and can be used in RTE systems. We identify some desirable characteristics of an alignment component for such applications by considering an ideal case, and the way reality departs from the ideal. We use these observations to assess previous work on alignment in the context of RTE, and to motivate our approach to alignment.

2.1 The Purpose of Alignment

The goal of alignment components is to decompose the text and hypothesis into semantic constituents, and determine which text constituent should be compared to which hypothesis constituent. However, there is more than one way in which such alignments are used: for example, to combine local decisions into a global decision ("alignment as entailment"); or as a means to determine which local comparisons should be made ("alignment as filter"). In the former case, a scoring function may be tuned and a threshold applied to determine entailment/non-entailment. In the latter case, alignment edges select among local decisions and are used to inform a separate subsequent step – for example, these local decisions may be used as features in one or more classifiers, or may guide the selective application of more expensive resources.

2.2 Desirable Characteristics of Alignment

The goal of alignment is to select among local decisions. In the ideal case, our alignment is maximally informed: that is, sufficient information is explicitly encoded in a way that allows alignment to select those local decisions that will result in the correct entailment decision.

Robust entailment recognition requires understanding of natural language text, and a suitable representation. If such a representation exists and is accurately encoded as constituents and structure in the text and hypothesis, and if appropriate metrics exist, these deep structures can be compared, scored, and modeled to determine the best alignment. In a variation of this ideal, the semantics of the representation is truly compositional, and we do not require a specific metric at the deepest structural level; instead, we use the deep structure to select among comparisons of smaller constituents, and apply the metrics for those constituents to generate the optimal alignment. This requires that metrics operate on the same scale, lest an incorrect alignment have a better score than a correct alignment.

In reality, we do not have truly deep structure, though we do have (noisy) analytics that operate at a number of granularities. Our metrics, which compare constituents from given analytic resources, are not guaranteed to generate outputs with comparable distributions (i.e., they may not be *compatible*).

In the light of desired and actual component behaviors, the following characteristics are desirable for alignment components:

- Alignment should simplify the entailment problem; a complete graph over text and hypothesis constituents is in some sense an alignment, but we are interested in identifying local comparisons that are directly relevant to the entailment decision (such as mapping an entity in the hypothesis to a corresponding entity in the text, instead of a function word). Alignments should therefore respect some constraints on the number of edges.
- Alignment should be robust to noise in individual resources; it should, if possible, avoid the limitations of a pipelined approach (and the attendant propagation of errors).
- Alignment should permit incorporation of metrics that are not mutually compatible (i.e., use different output scales and/or distributions of similarity values) as-is, but take advantage of those that *are* compatible.
- Given a new analytical resource, it should be easy to incorporate it into an existing alignment framework.
- Alignment should not require large amounts of labeled data for training.
- Alignment should accommodate constituents at multiple granularities.
- If an aligner has access to deep structure relevant to entailment decisions, it should use it and show improved performance. Performance should degrade gracefully as the reliability of the deep analysis decreases.

2.3 Previous Work

RTE researchers have used alignment in a number of ways, and in some cases, begun to define what alignment means in the context of entailment. This section discusses some representative examples.

The system described by (Zanzotto and Moschitti, 2006) can be framed in terms of the "alignment as entailment" approach, as they define an intra-pair alignment function and a distance metric between alignments for different pairs, then use the entailment labels to learn a separator based on a combination of these metrics. They propose an elegant model for cross-pair similarity based

on tree kernels applied to syntactic parse trees. They use a pipeline model in which similar pairs of terms in the text and hypothesis are replaced by placeholders, which are used to focus the structural similarity computations. However, simply adding more similarity resources is problematic as this will tend to increase the number of matching sites, and therefore increase the search space of the tree kernel alignment step. There is also the problem of accounting for small structural differences which have a disproportionate effect on cost – e.g. conditional and factive structures. Finally, as the similarity computation is based on syntactic structure, it is hard to envision a way to incorporate non-token-level analytical resources in an uncanonized fashion.

(Hickl et al., 2006; de Marneffe et al., 2007) follow the "alignment as filter" approach. (Hickl et al., 2006) use a pipelined approach to incorporate named entity and coreference information in the surface text. They use human annotated data to train a maximum entropy classifier to determine entailment of shallow parse chunks. The classifier uses features based on named entity information, WordNet similarity, and string edit distance (among others). Features from the resulting chunk-level alignment are used together with features from other sources to train a classifier that makes a global entailment decision. In addition to committing to a pipelined preprocessing system, it is not clear if their approach can be easily extended to incorporate new analytics, especially if they are not at the token or phrase level of representation.

(de Marneffe et al., 2007) investigate alignment more carefully, and formalize it as an optimization problem that accounts for alignments of individual tokens in the hypothesis and of pairs of hypothesis tokens connected by a dependency edge. They use human-annotated alignment data to train their aligner, which they evaluate in its own right. This is the basis of the alignment step in the entailment system described in (MacCartney et al., 2006), where it is used as a source of features for a global classifier. While they present a useful formulation of alignment in terms of an objective function, it accounts only for local structure in the hypothesis.

(MacCartney et al., 2008) generalize the alignment problem to the phrase level (where *phrase* simply means *contiguous text span*), and formalizes the alignment score in terms of equality, substitution, insertion, and deletion of phrases in the Text with respect to the Hypothesis. They train this model using lexical alignment labelings generated by (Brockett, 2007). While they report an improvement over two lexical-level alignment baselines, they do not observe significant difference in performance by the phrase-level system compared to a token-level alignment by the same system (i.e., where the phrase size is fixed at one token). One problem with this approach is that it appears to disregard known constituent boundaries and does not seem to offer a clean mechanism for applying specialized similarity resources in ways other than uniformly across all contiguous text spans. Moreover, it requires labeled alignment data, of which only a limited amount is available, and that too only at the token level.

2.4 Alignment for Feature Selection

There are (at least) three ways in which "Alignment as Filter" could inform feature selection for entailment decisions: 1. it can identify which constituents in the hypothesis match which constituents in the text, and inform a subsequent decision about the deep structure connecting the constituents; 2. it can use the deep structure to inform a set of subsequent decisions over local constituents (i.e. constrain the set of local comparisons using comparable deep structure); or 3. it can try to solve both problems simultaneously (similar to the "Alignment as Entailment" approach).

TEXT: John bought four books and three pencils when he visited the bookstore.
HYP 1: John went to the bookstore.
HYP 2: John has four pencils.

Figure 1. Example of two Textual Entailment pairs.

Consider the example in figure 1. Approach 1 might determine that "[John] [went to] [the bookstore]" from *HYP* 1 matches "[John] [visited] [the bookstore]" from *TEXT*. The entailment decision then requires that the structural connection of "John" and the term "visit" be detected or inferred. Approach 2 might match the predicate "[John] [has] [four pencils]" in *HYP* 2 with "[John] [bought] [four books and three pencils]" in *TEXT*; the entailment decision then hinges on the local decision as to whether "four pencils" matches "four books and three pencils". One instantiation of approach 3 involves considering all possible matches at both the deep structural level

and at the shallow level, and optimizing the alignment jointly over the two sets.

It is hard to imagine the third approach succeeding without requiring all metrics to be compatible. In the current work, we focus on a method intended to straddle approaches 1 and 2, but avoid the complications of approach 3.

3 Relation Alignment for Textual Entailment Recognition (RATER)

The RATER framework has four major components comparable to previous RTE systems:

1. **Preprocessor.** Annotates the entailment pair with a range of analytical tools.
2. **Graph Generator.** Applies metrics to constituents in specified annotation views to generate a match graph over the Text and Hypothesis constituents of the entailment pair.
3. **Aligner.** Filters the edges in the match graph to focus the feature extraction step.
4. **Feature Extractor/Classifier**. Extracts features based on the alignment output, and labels the input example.

The following sections describe these components and compare them to previous work.

3.1 Preprocessing and Data Representation

The preprocessing stage annotates the underlying text with tokenization and sentence splitting, part-of-speech (Roth and Zelenko, 1998), named entity (Ratinov and Roth, 2009), shallow- (Punyakanok and Roth, 2005) and syntactic-parse (Charniak and Johnson, 2005), semantic role labels (Punyakanok et al., 2008), multi-word expressions, phrasal verbs, coreference (Bengtson and Roth, 2008), modality, and quantifiers.

Rather than follow the pipelined approach of many RTE systems, we represent the combined analysis of the text in each entailment pair using the MRCS representation described in (Roth and Sammons, 2008), which comprises a set of stand-off annotations of the text. Each resource generates a separate view of the underlying text, or augments a view produced by another tool (specifically, modality and quantifiers augment the views generated by semantic role labelers). Each view is populated with *constituents* representing semantic components of a text span, identified by an analytical resource. Two constituents may be linked by an *edge* representing a structural relation. Figure 2 shows an example of this representation.

For example, a constituent from the named entity view represents a mention of an entity, while a constituent from the semantic relation view represents a relation (a predicate and its arguments, each a constituent in the semantic frame component view). Examples of *edges* representing structural relations between constituents include edges in dependency graphs and roles in semantic frames.

This representation is designed to facilitate comparison of Hypothesis and Text constituents from different views with each other, to minimize canonization, and to be easily extended to incorporate new analytical tools. The Meaning Representation described in the following section is lifted directly from the semantic role labeling views of the data representation.

3.2 Meaning (and Knowledge) Representation

Previous efforts to map natural language into canonical logical forms (such as (Bos and Markert, 2006)) have not been as successful as approaches working directly with the natural language representation, whether based purely on the lexical level (e.g. (Adams, 2006)) or using shallow induced structure (e.g. (Bar-Haim et al., 2007b)). Logical systems tend to be overly vulnerable to error propagation: for example, an error in determining the sense of a polysemous word directly affects unification of the induced logical form.

Approaches such as (Bar-Haim et al., 2007a) use natural language rules that map between sentence structures. The edges of these structures typically represent edges in a syntactic full- or dependency parse tree, while nodes are either natural language expressions or variables representing shared content between the mapped sentence structures. When determining whether a rule structure matches that of a given text span, lexical resources such as WordNet may be used to generalize the rule in a controlled way, greatly increasing expressiveness of individual rules. Typically, rules are applied to the text of the entailment pair in order to explicitly represent the implicit and entailed meanings of the underlying text, which may require chaining of rules by trying to apply each rule in the rule base over multiple iterations.

This ability to encode background knowledge

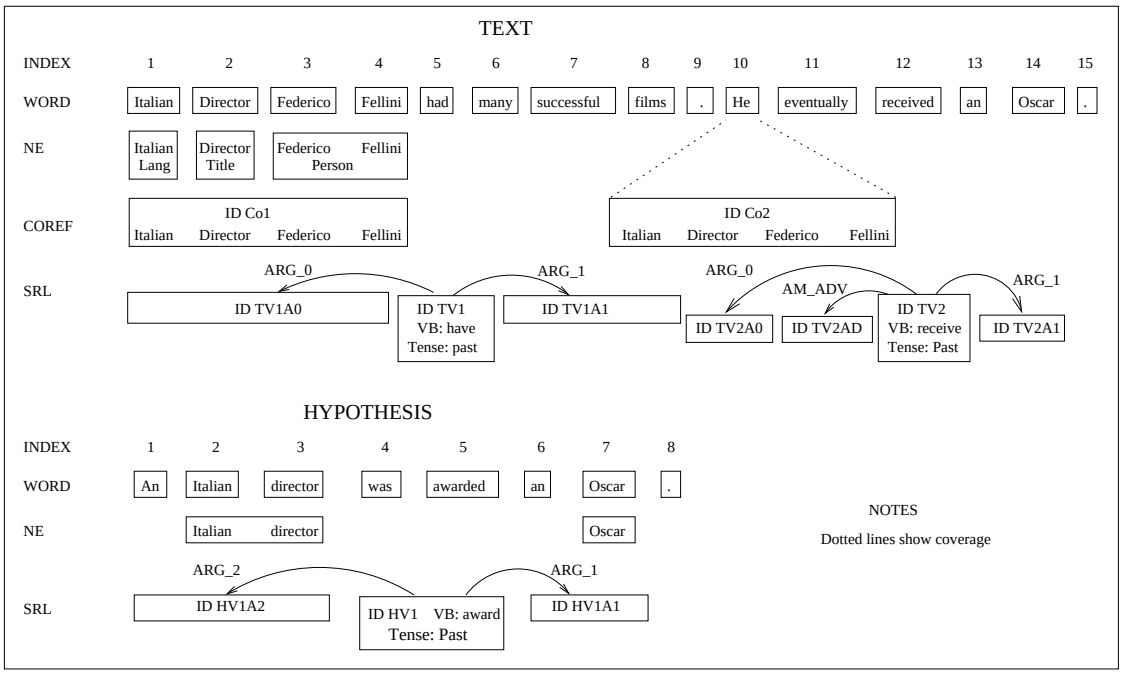

Figure 2. Example of a Textual Entailment pair as represented in MRCS.

in a representation that avoids the multiplicity of fully grounded natural language expressions, while to a large extent avoiding the limitations of canonical forms, is very appealing, though not without its drawbacks. In particular, background knowledge must still be acquired, and the expressiveness of the chosen natural language structure must be accounted for. Syntactic parses are relatively complex structures, and text spans with very similar lexical content may be expressed with a large number of syntactic structures – passive-active, light verb, and node raising constructions greatly complicate the process of generating rules with sufficient coverage to apply to more than a few sentences. Moreover, minor errors in attachments of constituents in parse trees may impact the effectiveness of such rules, and chaining of rules which may themselves be inaccurate can lead to inaccurate representations of meaning.

For these reasons, we favor shallow semantic structure based on the PropBank annotation standard (Kingsbury et al., 2002), a much flatter tree-like representation of text. Rather than focus on rule generation at this level of representation, we experiment with resources that abstract declarative knowledge into Metrics (section 3.3), and directly compare the shallow semantic structures themselves. We rely on enhancing the semantic structure, which may be achieved with rules or with other analytic resources, and on a learning framework that uses features informed by the semantic structure, but not constrained to a specified scoring function and threshold (section 3.7).

While we have experimented with rules mapping between SRL-based structures, the benefit of rules derived from automatically extracted paraphrase resources such as DIRT (Lin and Pantel, 2001) is limited by their specificity and their noise, while our relation comparison metric (which incorporates the lexical similarity metric based on WordNet, (Miller et al., 1990)) obviates the need for many simple rules. Rules derived from VerbNet (Kipper et al., 2000) generally map specific verbs to very general ones, and are therefore of limited value, in our experience, for the RTE task. However, such resources, if available, can be encapsulated as metrics or used by an annotator to augment existing views.

3.3 Comparing Semantic Units with Metrics

Determining whether the Hypothesis is entailed by the Text depends heavily on local entailment decisions, such as recognizing when two mentions of an entity refer to the same underlying referent, and identifying when a word in the text has the same meaning as a given word in the hypothesis. As described in section 2, many such local decisions may inform the final entailment decision.

Often, there are multiple matching candidates in the text for a given term in the hypothesis. In

order to compare different mappings, it is necessary to define relevance or similarity of individual constituents to other constituents, and this is made easier when such comparisons may be ranked. To this end, we encapsulate declarative knowledge in Metrics where possible.

An Entailment Metric is defined as a resource that compares two semantic constituents of specific types and returns a real number in the range $[-1, 1]$. A score close to 1 indicates high similarity, close to 0 means irrelevant, and close to -1 indicates contradiction (e.g. antonymy). High magnitude therefore indicates strong relevance, while low magnitute indicates weak relevance.

Metrics may be defined over arbitrary constituents, and may compose results from metrics over sub-constituents. We have implemented metrics for comparing named entities, words and multi-word phrases, numerical quantities, and semantic relations. Metrics can encapsulate rule-based resources as well as classifiers, and therefore have the potential to generalize over different approaches to RTE.

The Graph Generator applies our metrics to appropriate semantic constituents in the multi-view representation of the entailment pair, representing the result of each such comparison with an edge whose weight is the score assigned by the metric that generated it.

3.4 Aligning Natural Language Relations

We wish to use the alignment step to filter the edges in the Graph Generator output to focus feature extraction. The aligner must be able to integrate different information sources at a variety of granularities, take advantage of deep structure annotation where available, yet degrade gracefully when it is not.

We resolve these problems by formulating *multiple* alignments, each of which integrates information from comparable resources that have compatible metrics. This allows comparable resources (for example, multi-word expressions and individual words, both of which use WordNet-derived metrics) to compete with each other, but separates metrics with different scales (such as Named Entity metrics, which are based on rules and string similarity measures). To ensure that the alignment is informative, we constrain it by requiring that each token in the hypothesis be aligned at most once, i.e. only a single constituent covering a given token may be selected from the set of all constituents covering that token.

We then extract features from each alignment, and from comparisons between different alignments over the same entailment pair. In particular, we compare the natural language relation alignment with other views, with the intuition that if the relations are successfully aligned, other views provide relevant information. For example, if the named entity alignment disagrees with the relation alignment, this may suggest that the relation alignment is incorrect.

We use the feature extraction approach to provide robustness against analytic and metric noise and incompleteness of relation annotation resources. If deeper structure is successfully annotated and the relevant metrics are good, these deep features should be strong. If the deep structure resources are too noisy or absent, features from other views/alignments will become more important.

3.5 Constrained Optimization Model for Alignment-As-Filter

Alignment is formulated as an optimization problem subject to the constraint that each token in the hypothesis must be mapped to at most one target in the text, so constituents covering more than one token may not overlap. Constituents at different granularities may both have alignment edges in an optimal solution, provided they do not overlap.

Since metrics may return negative scores, the objective function must account for these. Negative scores indicate contradiction: in the absence of a better positive match, this information may be highly relevant to the subsequent entailment decision. In the objective function, therefore, the magnitude of the edge weight is used. The edge retains a label indicating its negativity, which is used in the feature extraction stage.

For alignments over shallow constituents, we must guess at the deep structure; we therefore include locality in the objective function by penalizing alignments where neighboring constituents in the hypothesis are paired with widely separated constituents in the text. We ignore crossing edges, as we do not believe these are reliably informative of entailment.

The objective function is then:

$$\frac{\sum_i e(H_i, T_j) + \alpha . \sum_i \Delta(e(H_i, T_j), e(H_{i+1}, T_k))}{m} \quad (1)$$

and the constraint:

$$\sum_j I[e(H_i, T_j)] \leq 1 \quad (2)$$

where m is the number of tokens in the hypothesis; $e(H_i, T_j)$ is the magnitude of the score of a metric comparing hypothesis token i and text token j; and α is a parameter weighting the distance penalty. $\Delta(e(H_i, T_j), e(H_{i+1}, T_k))$ measures the distance, in tokens, between the text constituent aligned to hypothesis token i and the text constituent aligned to hypothesis token $i+1$. For constituents covering multiple tokens, this value is the *minimum* distance between any token covered by the constituent covering T_j and any token covered by T_k. $I[e(H_i, T_j)]$ is an indicator function indicating that token i in the hypothesis is mapped to token j in the text. Note that this differs from the formulation in (de Marneffe et al., 2007), as it accounts for the locality of mapped constituents in the text in addition to that in the hypothesis. Note also that distance could be defined over structures like dependency trees, although we have not yet investigated such options.

For alignments that combine constituents of different granularities, the formulation above uses as token-level edge-weights the magnitude of the edge score for the mapped constituents covering the pair of tokens in question. Note that if constituents from different views are combined in this way, their metrics must be compatible.

Since we did not have training data, we selected the alignment parameter α by hand (a positive value close to zero, sufficient to break ties), and used brute force search to find the optimal alignment. The search time has an upper limit, after which a greedy left-to-right alignment is used in place of the optimal solution.

3.6 Constrained Optimization Model for Alignment-As-Entailment

We also performed experiments with the "alignment as entailment" approach, as described in (Chang et al., 2010). In this setting, we represent each of the candidate sentences as a unified graph consisting of SRL, dependency parse, NER, and coreference views (we used the coreference view to substitute canonical mentions for pronouns). We scored candidate pairs based on our ability to align the constituents of these views.

The constituents we considered were individual tokens and named entities, and the relations indicated SRL argument types (between verbs and the heads of their arguments) and dependency edges (within SRL arguments that spanned more than one token or named entity). The alignment process was framed as a constrained optimization process that picks the best matching for constituents under constraints forcing the validity of the resulting alignment.

Conventional alignment methods weigh local alignment decisions according to external similarity metrics, and ignore existing labeled data, using it only when optimizing the classification model. Our approach aims to optimize the alignment model parameters using the labeled data. In this setting, the problem of learning the entailment classifier is not separate from that of aligning the graphs. We use the aligned constituents as features for learning an entailment model using labeled data, and propagate the learned feature weights to the aligner, where they can be used to weigh competing alignment decisions. We iterate over this model until the weights converge.

Formally, we optimize the following objective function, when performing alignment :

$$\sum_i W^T \phi(e(H_i, T_j)) + \sum_i W^T \phi(e(r_i(H_j, H_k), r(T_l, T_m)))$$

where W is a weight vector, and ϕ is a feature mapping defined over the chosen constituent alignments. The formula is separated into two parts: the first considers features extracted from token alignments (words and named entities), and is denoted as $e(H_i, T_j)$, where H_i, T_j correspond to the mapped tokens in hypothesis and text, respectively. The second part is defined over relations (SRL and dependency parse) and is denoted as $e(r_i(H_j, H_k), r(T_l, T_m))$, where $r(\cdot, \cdot)$ denotes a directed, labeled edge between tokens. In addition, we enforce consistency between alignment of these two views by adding the following constraint :

$$\forall j, k, l, m \;\; e(r(H_j, H_k), r(T_l, T_m)) \Leftrightarrow e(H_j, T_l) \wedge e(H_k, T_m)$$

The alignment process can now be simply explained by the procedure outlined in figure 3.

The results of this approach are provided in table 2 under the heading *Alignment as Entailment*.

Feature Name #	Type	Description
Word-LLM	real	Sum of scores of word edges selected by aligner, averaged by number of tokens in hypothesis
Frac-TokenMatch	real	Fraction of words in H that are aligned to T with non-zero edges.
Frac-NEmatch	real	Fraction of NEs in H that have a non-zero edge to some NE token in T (using NESim metric)
SRL-verbLLM	real	Consider only SRL verbs (over all SRL relations in H), and find LLM using the Word-view edges for those words
SRL-coreLLM	real	Same as **SRL-verbLLM**, but for all core arguments (ARG0,1,2) (includes all relations in H)
SRL-verbScore	real	Sum of scores of edges that align SRL verbs in H to some SRL argument in T, averaged by number of relations in H: this uses SRL view, when present
SRL-coreScore	real	Same as **SRL-verbScore**, but for core SRL arguments (ARG0,1,2)
SRL-verb-none	boolean	Active if, in the SRL view, the verb did not align to any argument
SRL-verb-sth	boolean	Active if, in the SRL view, the verb aligned to some argument (verb or A*)
SRL-core-none	boolean	Active if, in the SRL view, none of the core arguments (ARG0,1,2) aligned to any argument
SRL-core-some	boolean	Active if, in the SRL view, some but not all core arguments (ARG0,1,2) aligned to some argument
SRL-core-all	boolean	Active if, in the SRL view, all core arguments (ARG0,1,2) aligned to some argument
SRLpos-core-Bef	boolean	Active if all core arguments before the verb in H are aligned to arguments before the verb in T
SRLpos-core-Aft	boolean	Active if all core arguments after the verb in H are aligned to arguments after the verb in T
VB-none	boolean	Active if VERB tokens did not match any token in T
VB-match	boolean	Active if VERB tokens match some arguments in T (no SRL constraint check)
Acore-none	boolean	Active if tokens from none of the core arguments ARG0,1,2 match T
Acore-some-cons	boolean	Active if tokens from some (but not all) core arguments ARG0,1,2 match "consecutive" tokens in T
Acore-all-cons	boolean	Active if tokens from all core arguments ARG0,1,2 match "consecutive" tokens in T
Acore-some-match	boolean	Active if tokens from some (but not all) core arguments ARG0,1,2 match "consecutive" tokens in T
Acore-all-match	boolean	Active if tokens from all core arguments ARG0,1,2 match some arguments in T (no SRL constraint check)
Con-root	boolean	Active if any contradiction feature is active (negation, poor predicate relation match, poor NE match)

Table 1. Features used in the RATER classifier

while $\neg$ *converged* **do**
  $D = \emptyset$
  for all (T,H) $\in$ Training Data **do**
    $D = D \cup ALIGN_W(T, H)$
  end for
  $W = LEARN(D)$
end while

Figure 3. Alignment Learning Algorithm

3.7 Learning and Classifying

Since we did not have a large amount of training data, we kept the number of alignment-based features relatively low to keep the learning problem simple. We intended the features to be intuitive: they should make sense from the perspective of the general problem of entailment, not just in the context of this particular data set. The feature types are presented in table 1.

We trained an SVM classifier using a slightly adapted version of (Fan et al., 2008). The classifier was trained on the RTE5 development set.

4 Evaluation and Discussion

To evaluate the effectiveness of the RATER approach, we performed an ablation study on our RTE system. The system should show improvement with additional (informative) resources, should not overfit to the training set, and its performance should improve if reliable deep structure cues are present.

A significant problem for most analytical resources is that they are intra-sentence; at present, the only inter-sentence analytic resource available to us is our coreference resolver. To test the system's response to deep structure cues, we integrated this with the shallow semantic predicate-argument structures by adding coreferent mentions of arguments in the predicate-argument structures as additional arguments with the same role, and retraining the classifier on the resulting alignment output.

	RTE5 Dev				RTE5 Test			
System Version #	**Overall**	**QA**	**IE**	**IR**	**Overall**	**QA**	**IE**	**IR**
Baseline	**0.628**	0.641	0.557	0.683	**0.600**	0.550	0.500	0.750
Submitted Run *	**0.630**	0.632	0.537	0.689	**0.643**	0.585	0.595	0.750
Without NE *	**0.627**	0.610	0.579	0.688	**0.595**	0.565	0.515	0.705
Submitted Run (fixed)	**0.648**	0.647	0.552	0.744	**0.644**	0.580	0.576	0.775
Without NE (fixed)	**0.640**	0.631	0.577	0.708	**0.629**	0.580	0.530	0.775
Simple NE	**0.623**	0.655	0.543	0.670	**0.633**	0.580	0.605	0.715
Without WN	**0.647**	0.650	0.533	0.755	**0.603**	0.565	0.535	0.710
Submitted Plus Coref	**0.663**	0.665	0.559	0.765	**0.666**	0.596	0.615	0.785
Alignment As Entailment	**0.667**	0.645	0.555	0.800	**0.670**	0.640	0.540	0.830

Table 2. Performance of different versions of the system. NE means "Named Entity"; WN means "WordNet".

We also trained and evaluated a baseline version of the system by deactivating all except the lexical alignment features.

Table 2 compares the results of the different versions of our system on the RTE5 Dev and Test corpora, and includes the ablation runs we submitted. The learning component of the system was trained on the RTE5 Development set, so the 'RTE5 Dev' results in the table show a self-training bias.

On examining the trace files for two of the runs we submitted originally (marked with an asterisk in table 2), we observed errors arising from a bug in our system that affected the word-level alignments of approximately 60 development examples and 90 test examples. We re-ran the corrected versions of the system for those examples, to get the results marked "fixed" in the table of results.

The ablation results indicate the importance of both Named Entity recognition/resolution and WordNet-based similarity to our system. In our framework, WordNet mappings are especially useful when they allow relation predicates in the text and Hypothesis to be matched.

The breakdown by task shows that although it is beneficial in the general case, the more complex NE actually hurts performance in the IE subtask. The reason is that IE examples often lack explicit structures (from the system's perspective) and so decisions are made based more on similarity of Hypothesis constituents to Text constituents. The simpler system generates different scores for non-identical Named Entity pairs; for example, in test example 4, the simple metric assigns a similarity score of zero to the pair (Santana, Juan Carmelo Santana), while the more advanced metric assigns it a score of 1.0. This leads to different feature weights, and to different classifications, e.g. test example 15.

The benefit of additional deep structure is apparent from the "Submitted Plus Coref(erence)" results. When this additional structure is explicitly encoded, more positive examples have comparable predicate-argument structures, resulting in stronger cues from agreement between predicate-argument alignments and those of other views, and hence more reliable features in the final classifier.

In general, the results support our intuitions that the RATER approach we have outlined in this work has the desired attributes specified in section 2.2: new resources can be added in a modular fashion via annotators and metrics, and have a significant effect on system performance; and making implicit structure explicit in terms of predicate-argument structures improves system performance as expected. Overall, the results show good generalization of the system trained on the development data when evaluated on the test data; this makes sense, as all the resources we add are intended to be generally useful for NLP applications, and are not tuned for or developed from the RTE5 data.

5 Conclusions and Future Work

We have identified three ways in which alignment may be used in relation to making entailment decisions, and identified some desirable characteristics for alignment components. We have implemented a system based on this analysis, designed to handle non-scaled comparison resources and different analytic granularities, and to have a graceful degradation in performance if some resources are unavailable or unreliable for a particular domain. Our experimental evaluation on the RTE5 data indicates that our effort has been reasonably successful.

We hope that our work contributes to the understanding needed for the development of a general entailment framework; we plan to improve and ultimately release the code for our data representation and comparison resources.

We are presently investigating ways to learn the

parameters for the existing alignment component, and to improve the alignment-as-entailment approach.

Acknowledgments

This work is funded by a grant from Boeing; by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC; and with support from DARPA under the Machine Reading program.

References

[Adams2006] Rod Adams. 2006. Textual entailment through extended lexical overlap. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

[Bar-Haim et al.2007a] Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007a. Semantic inference at the lexical-syntactic level. In *Proceedings of the AAAI*, Vancouver, July. AAAI.

[Bar-Haim et al.2007b] Roy Bar-Haim, Ido Dagan, Iddo Greental, Idan Szpektor, and Moshe Friedman. 2007b. Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 131–136, Prague, June. Association for Computational Linguistics.

[Bengtson and Roth2008] E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. pages xx–yy, Oct.

[Bos and Markert2006] Johan Bos and Katja Markert. 2006. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

[Brockett2007] C. Brockett. 2007. Aligning the rte 2006 corpus. Technical Report MSR-TR-2007-77, Microsoft Research.

[Chang et al.2010] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *NAACL*.

[Charniak and Johnson2005] Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. of the Annual Meeting of the ACL*, pages 173–180, Ann Arbor, Michigan. ACL.

[de Marneffe et al.2007] Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Cer, Daniel Ramage, Chloe Kiddon, and Christopher D. Manning. 2007. Aligning semantic graphs for textual inference and machine reading. In *AAAI Spring Symposium at Stanford 2007*.

[Fan et al.2008] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

[Hickl et al.2006] Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with lcc's groundhog system. In *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*.

[Kingsbury et al.2002] P. Kingsbury, M. Palmer, and M. Marcus. 2002. Adding semantic annotation to the Penn treebank. In *Proceedings of the 2002 Human Language Technology conference (HLT).*, San Diego, CA.

[Kipper et al.2000] Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX. AAAI.

[Lin and Pantel2001] D. Lin and P. Pantel. 2001. DIRT: discovery of inference rules from text. In *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001*, pages 323–328.

[MacCartney et al.2006] B. MacCartney, T. Grenager, and M. de Marneffe. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of RTE-NAACL 2006*.

[MacCartney et al.2008] Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2008)*.

[Miller et al.1990] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

[Punyakanok and Roth2005] V. Punyakanok and D. Roth. 2005. Inference with classifiers: The phrase identification problem. *Computational Linguistics*. In submission.

[Punyakanok et al.2008] V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

[Ratinov and Roth2009] L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, Jun.

[Roth and Sammons2008] D. Roth and M. Sammons. 2008. A unified representation and inference paradigm for natural language processing. Technical Report UIUCDCS-R-2008-2969, UIUC Computer Science Department, Jun.

[Roth and Zelenko1998] D. Roth and D. Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING-ACL, The 17th International Conference on Computational Linguistics*, pages 1136–1142.

[Zanzotto and Moschitti2006] Fabio Massimo Zanzotto and Alessandro Moschitti. 2006. Automatic learning of textual entailments with cross-pair similarities. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 401–408, Sydney, Australia, July. Association for Computational Linguistics.