Feature-Rich Two-Stage Logistic Regression for Monolingual Alignment

Md Arafat Sultan[†], **Steven Bethard**[‡] **and Tamara Sumner**[†]
[†]Institute of Cognitive Science and Department of Computer Science
University of Colorado Boulder
[‡]Department of Computer and Information Sciences
University of Alabama at Birmingham
arafat.sultan@colorado.edu, bethard@cis.uab.edu, sumner@colorado.edu

Abstract

Monolingual alignment is the task of pairing semantically similar units from two pieces of text. We report a top-performing supervised aligner that operates on short text snippets. We employ a large feature set to (1) encode similarities among semantic units (words and named entities) in context, and (2) address cooperation and competition for alignment among units in the same snippet. These features are deployed in a two-stage logistic regression framework for alignment. On two benchmark data sets, our aligner achieves F_1 scores of 92.1% and 88.5%, with statistically significant error reductions of 4.8% and 7.3% over the previous best aligner. It produces top results in extrinsic evaluation as well.

1 Introduction

Computer applications frequently require semantic comparison between short snippets of natural language text. Such comparisons are key to paraphrase detection (Das and Smith, 2009; Madnani et al., 2012), textual similarity identification (Agirre et al., 2015; Sultan et al., 2015) and recognition of textual entailment (Dagan and Glickman, 2004; Padó et al., 2015). And they underpin applications such as short answer grading (Mohler et al., 2011), question answering (Hixon et al., 2015), machine translation evaluation (Padó et al., 2009), and machine reading (de Marneffe et al., 2007).

A central problem underlying all text comparison tasks is that of *alignment*: pairing related semantic units (i.e. words and phrases) across the two snippets (MacCartney et al., 2008; Thadani and McKeown, 2011; Thadani et al., 2012; Yao et al., 2013a; Yao et al., 2013b; Sultan et al., 2014a). Studies have shown that such tasks can benefit from an explicit alignment component (Hickl and Bensley, 2007; Sultan et al., 2014b; Sultan et al., 2015). However, alignment is still an open research problem. We present a supervised monolingual aligner that produces top results in several intrinsic and extrinsic evaluation experiments. We pinpoint a set of key challenges for alignment and design a model with components targeted at each.

Lexical and phrasal alignments can both be represented as pairs of words – in the form of many-to-many mappings among the two phrases' component words in the latter case. Thus without loss of generality, we formulate alignment as a binary classification task where given all word pairs across two sentences, the goal is to assign each a class label in {*aligned*, *not aligned*}. However, this is not a straightforward classification scenario where each word pair can be treated independently – words in the same snippet can play both mutually cooperating and competing roles in complex ways. For example, semantically similar words in a snippet can be in competition for alignment with a word in the other snippet, whereas words that constitute a phrase can provide supporting evidence for one another (e.g. in named entity alignments such as `Watson` ↔ `John Hamish Watson`). To handle such interdependencies, we employ a two-stage logistic regression model – stage 1 computes an alignment probability for each word pair based solely on its own feature values, and stage 2 assigns the eventual alignment labels to all pairs following a comparative assessment of stage 1 probabilities of cooperating and competing pairs.

On two alignment data sets reported in (Brockett, 2007) and (Thadani et al., 2012), our aligner demonstrates respective F_1 scores of 92.1% and 88.5%, with statistically significant error reductions of 4.8% and 7.3% over the previous best aligner (Sultan et al., 2014a). We also present extrinsic evaluation of the aligner within two text comparison tasks, namely sentence similarity identification and paraphrase detection, where it demonstrates state-of-the-art results.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 949–959,
Lisbon, Portugal, 17-21 September 2015.

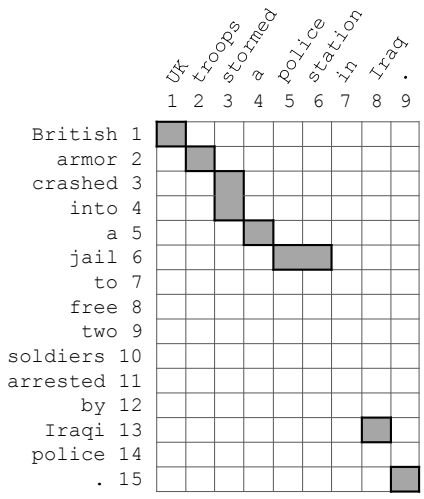

Figure 1: A human-aligned sentence pair from the MSR alignment corpus. The shaded cells depict the alignment, which can also be represented as the set of word index pairs $\{(1,1),(2,2),(3,3),(4,3),(5,4),\ldots,(15,9)\}$.

2 Alignment: Key Pieces of the Puzzle

We illustrate with examples key pieces of the alignment puzzle and discuss techniques used by existing aligners to solve them. We use the term 'unit' to refer to both words and phrases in a snippet.

Figure 1 shows a shortened version of sentence pair 712 in the MSR alignment corpus *dev* set (Brockett, 2007) with related units aligned by human annotators. Evident from these alignments is the fact that aligned units are typically semantically similar or related. Existing aligners utilize a variety of resources and techniques for computing similarity between units: WordNet (MacCartney et al., 2008; Thadani and McKeown, 2011), PPDB (Yao et al., 2013b; Sultan et al., 2014a), distributional similarity measures (MacCartney et al., 2008; Yao et al., 2013b) and string similarity measures (MacCartney et al., 2008; Yao et al., 2013a). Recent work on neural word embeddings (Mikolov et al., 2013; Baroni et al., 2014) have advanced the state of distributional similarity, but remain largely unexplored in the context of alignment.

Lexical or phrasal similarity does not entail alignment, however. Consider function words: the alignment $(5,4)$ in Figure 1 exists not just because both units are the word `a`, but also because they modify semantically equivalent units: `jail` and `police station`. The influence of context on content word alignment becomes salient particularly in the presence of competing words. In Figure 1, `(soldiers(10), troops(2))` are not aligned despite the two words' semantic equivalence *in isolation*, due to the presence of a competing pair, `(armor(2), troops(2))`, which is a better fit *in context*.

The above examples reveal a second aligner requirement: the ability to incorporate context into similarity calculations. Existing supervised aligners use various contextual features within a learning algorithm for this purpose. Such features include both shallow surface measures (e.g., the relative positions of the tokens being aligned in the respective sentences, similarities in the immediate left or right words) (MacCartney et al., 2008; Thadani and McKeown, 2011; Yao et al., 2013a) and syntactic measures like typed dependencies (Thadani and McKeown, 2011; Thadani et al., 2012). Sultan et al. (2014a) design an unsupervised model that more directly encodes context, via surface and dependency-based neighbors which allow contextual similarity to be represented as a weighted sum of lexical similarity. But their model lacks a key structural advantage of supervised models: to be able to use an arbitrarily large feature set to robustly encode lexical and/or contextual similarity.

The third and final key component of an aligner is a mechanism to combine lexical/phrasal and contextual similarities to produce alignments. This task is non-trivial due to the presence of cooperating and competing units. We first discuss competing units: semantically similar units in one snippet, each of which is a potential candidate for alignment with one or more units in the other snippet. At least three different possible scenarios of varying difficulty exist concerning such units:

- Scenario 1: No competing units. In Figure 1, the aligned pair `(British(1), UK(1))` represents this scenario.
- Scenario 2: Many-to-one competition: when multiple units in one snippet are similar to a single unit in the other snippet. In Figure 1, pairs `(armors(2), troops(2))` and `(soldiers(10), troops(2))` are in such competition.
- Scenario 3: Many-to-many competition: when similar units in one snippet have multiple potential alignments in the other snippet.

Groups of mutually cooperating units can also exist where one unit provides supporting evidence

for the alignment of other units in the group. Examples (besides named entities) include individual words in one snippet that are grouped together in the other snippet (e.g., `state of the art` ↔ `state-of-the-art` or `headquarters in Paris` ↔ `Paris-based`).

We briefly discuss the working principles of existing aligners to show how they respsond to these challenges. MacCartney et al. (2008), Thadani and McKeown (2011) and Thadani et al. (2012) frame alignment as a set of phrase edit (insertion, deletion and substitution) operations that transform one snippet into the other. Each edit operation is scored as a weighted sum of feature values (including lexical and contextual similarity features), and an optimal set of edits is computed. Yao et al. (2013a; 2013b) take a sequence labeling approach: input snippets are considered sequences of units and for each unit in one snippet, units in the other snippet are considered potential labels. A first order conditional random field is used for prediction. Sultan et al. (2014a) treat alignment as a bipartite matching problem and use a greedy algorithm to perform one-to-one word alignment. A weighted sum of two words' lexical and contextual similarities serves as the pair's edge weight.

Noticeable in the designs of the supervised aligners is a lack of attention to the scenarios competing units can pose – alignment of a unit depends only on its own feature values. While the unsupervised aligner by Sultan et al. (2014a) employs techniques to deal with such scenarios, it allows only one-to-one alignment, which fundamentally limits the set of reachable alignments.

3 Approach

We primarily focus on word alignment, which Yao et al. (2013b) report to cover more than 95% of all alignments in multiple human-annotated corpora. Named entities are the only phrasal units we consider for alignment; in a later section we discuss how our techniques can be extended to general phrasal alignment.

Figure 2 shows our two-stage logistic regression model. We address the first two challenges, namely identifying lexical and contextual similarities, in stage 1 of the model. Given input text snippets $\boldsymbol{T}^{(1)} = (T_1^{(1)}, ..., T_n^{(1)})$ and $\boldsymbol{T}^{(2)} = (T_1^{(2)}, ..., T_m^{(2)})$ where $T_k^{(t)}$ is the k-th word of snippet $\boldsymbol{T}^{(t)}$, the goal of this stage is to assign each word pair of the form $(T_i^{(1)}, T_j^{(2)})$ an alignment

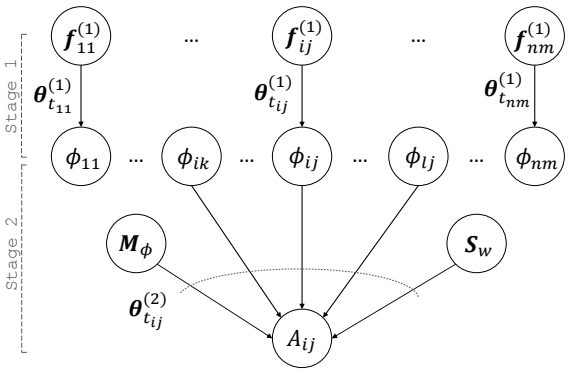

Figure 2: Two-stage logistic regression for alignment. Stage 1 computes an alignment probability ϕ_{ij} for each word pair based on local features $\boldsymbol{f}_{ij}^{(1)}$ and learned weights $\boldsymbol{\theta}_{t_{ij}}^{(1)}$ (see Section 4.1). Stage 2 assigns each pair a label $A_{ij} \in \{$*aligned*, *not aligned*$\}$ based on its own ϕ, the ϕ of its cooperating and competing pairs, a max-weighted bipartite matching $\boldsymbol{M}_\phi$ with all ϕ values as edge weights, the semantic similarities $\boldsymbol{S}_w$ of the pair's words and words in all cooperating pairs, and learned weights $\boldsymbol{\theta}_{t_{ij}}^{(2)}$ for these global features.

probability ϕ_{ij}, based on the pair's lexical and contextual similarity features. We discuss our stage 1 features in Section 4.1.

We categorize each word along two different dimensions: (1) whether or not it is part of a named entity, and (2) which of the following groups it belongs to: content words, function words, and punctuation marks. This distinction is important because, (1) certain features apply only to certain types of words (e.g., acronymy applies only to named entities; punctuation marks do not participate in dependency relationships), and (2) certain features can be more important for certain types of words (e.g., the role of a function word depends heavily on its surrounding words and therefore contextual features can be more important for function words). Combined, the two above dimensions form a domain of six possible values which can be represented as the Cartesian product $\{$*non-named entity*, *named entity*$\} \times \{$*content word*, *function word*, *punctuation mark*$\}$. Each member of this set is a *word type* in our model; for instance, *named entity function word* is a word type.

This notion of types is then extended to word pairs in $\boldsymbol{T}^{(1)} \times \boldsymbol{T}^{(2)}$: the type of pair $(T_i^{(1)}, T_j^{(2)})$ is the union of the types of $T_i^{(1)}$ and $T_j^{(2)}$. Given

the pair's stage 1 feature vector $\boldsymbol{f}_{ij}^{(1)}$ and the stage 1 weight vector $\boldsymbol{\theta}_{t_{ij}}^{(1)}$ for its type t_{ij}, we compute its stage 1 alignment probability ϕ_{ij} as:

$$\phi_{ij} = \frac{1}{1 + e^{-\boldsymbol{\theta}_{t_{ij}}^{(1)} \cdot \boldsymbol{f}_{ij}^{(1)}}}$$

The weight vector $\boldsymbol{\theta}_t^{(1)}$ for word pair type t is derived by minimizing the L1-regularized loss:

$$J(\boldsymbol{\theta}_t^{(1)}) = -\frac{1}{N_t} \sum_{p=1}^{N_t} \Big[y_t^{(p)} \log(\phi_t^{(p)}) + (1 - y_t^{(p)}) \log(1 - \phi_t^{(p)}) \Big] + \lambda \|\boldsymbol{\theta}_t^{(1)}\|_1$$

where N_t is the number of word pairs of type t over all sentence pairs in the training data, $y_t^{(p)}$ is the gold label for pair p of type t ($1 =$ *aligned*, $0 =$ *not aligned*), and $\phi_t^{(p)}$ is its stage 1 alignment probability.

Stage 2 of the model assigns the final alignment label $A_{ij} \in \{0, 1\}$ to $(T_i^{(1)}, T_j^{(2)})$. Like stage 1, it uses L1-regularized logistic regression to compute an alignment probability for each word pair, but additionally assigns a final $0/1$ label using a 0.5 threshold. Stage 2 factors in the stage 1 probabilities of cooperating and competing pairs as well as a maximum-weighted matching $\boldsymbol{M}_\phi$ between $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$, where word pairs in $\boldsymbol{T}^{(1)} \times \boldsymbol{T}^{(2)}$ are weighted by their stage 1 ϕ values. Such global knowledge is useful in addressing cooperation and competition among words. We describe our stage 2 features in Section 4.2.

The two stages are trained separately, each as *n* standard logistic regression models where *n* is the number of word pair types for which at least one instance per class is observed in the training data. The stage 1 models are first trained and used to make predictions for each training sentence pair (for each training pair, all other training pairs are used to train the model). Given all the stage 1 alignment probabilities and the other stage 2 features, the stage 2 models are then trained. At test time, the two sets of trained models (i.e. stage 1 and 2 models) are successively applied to each input sentence pair.

4 Features

As mentioned above, we train a separate model for each individual word pair type. Our feature set is largely the same across word pair types, with some differences. In the two following sections, we discuss these features and indicate the associated word pair types. We assume alignment of the two words $T_i^{(1)} \in \boldsymbol{T}^{(1)}$ and $T_j^{(2)} \in \boldsymbol{T}^{(2)}$.

4.1 Stage 1: Assessing Pairs Individually

4.1.1 Word Similarity Features

Our first feature combines neural word embeddings, used previously for word similarity prediction (Mikolov et al., 2013; Baroni et al., 2014), with a paraphrase database (Ganitkevitch et al., 2013). Our feature is the output of a ridge regression model trained on human annotations of word similarity (Radinsky et al., 2011; Halawi et al., 2012; Bruni et al., 2014) with two features: the cosine similarity between the neural embedding vectors of the two words (using a publicly available set of 400-dimensional word vectors (Baroni et al., 2014)), and the presence/absence of the word pair in the PPDB XXXL database. This regression model produces similarities (sim henceforth) in $[0, 1]$, though we only consider similarities above 0.5 as lower scores are often noisy. To deal with single-letter spelling errors, we consider $T_i^{(1)}$ and $T_j^{(2)}$ to be an exact match if exactly one of the two is correctly spelled and their Levenshtein distance is 1 (words of length ≥ 3 only).

We also use the following semantic and string similarity features: a boolean feature that is 1 iff one of $T_i^{(1)}$ and $T_j^{(2)}$ is hyphenated and the other is identical to a hyphen-delimited part of the first, the same feature for highly similar ($sim \geq 0.9$) words, two features that show what proportion of the characters of one word is covered by the other if the latter is a prefix or a suffix of the former and zero otherwise (length < 3 words are discarded).

For named entities, we (1) consider acronymy as exact match, (2) use membership in two lists of alternative country names and country-nationality pairs (from Wikipedia) as features, and (3) include a feature that encodes whether $T_i^{(1)}$ and $T_j^{(2)}$ belong to the same named entity (determined by one mention containing all words of the other, e.g., `Einstein` and `Albert Einstein`).

4.1.2 Contextual Features

Effective identification of contextual similarity calls for a robust representation of word context in a sentence. Our contextual features are based on two different sentence representations. The word-

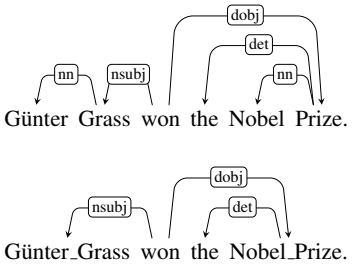

Figure 3: Word and entity-based representations of a sentence. Words in the same named entity are grouped together in the latter representation.

based representation treats each individual word as a semantic unit whereas the entity-based representation (1) groups together words in a multi-word named entity, and (2) treats non-name words as individual entities. Figure 3 shows an example. The two representations are complementary – the entity-based representation can capture equivalences between mentions of different lengths of a named entity, while the word-based representation allows the use of similarity resources for named entity words. Non-name words are treated identically. For simplicity we only discuss our word-based features below, but each feature also has an entity-based variant.

Dependency-based context. These features apply only if neither of $T_i^{(1)}$ and $T_j^{(2)}$ is a punctuation mark. We compute the proportion of identical and highly similar ($sim \geq 0.9$) parents and children of $T_i^{(1)}$ and $T_j^{(2)}$ in the dependency trees of $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$ (Stanford collapsed dependencies (de Marneffe et al., 2006)). Equivalent dependency types (Sultan et al., 2014a) are included in the above computation, which encode semantic equivalences between typed dependencies (e.g., *nsubjpass* and *dobj*). We employ separate features for identicality and similarity. Similar features are also computed for a dependency neighborhood of size 2 (parents, grandparents, children and grandchildren), where we consider only content word neighbors.

Dependency neighbors of $T_i^{(1)}$ and $T_j^{(2)}$ that are less similar ($0.9 > sim \geq 0.5$; e.g., (`gas, energy`) or (`award, winner`)) can also contain useful semantic information for an aligner. To accommodate this relatively large range of word similarities, rather than counting such pairs, we find a maximum-weighted bipartite matching of $T_i^{(1)}$ and $T_j^{(2)}$ neighbors in a neighborhood of size 2 using the primal-dual algorithm (content words only), where word similarities across the two neighborhoods serve as edge weights. We use as a feature the sum of similarities between the matched neighbors, normalized by the total number of content words in the two neighborhoods.

Surface-form context. We draw several contextual features from nearby words of $T_i^{(1)}$ and $T_j^{(2)}$ in the surface forms of $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$: (1) whether the left and/or the right word/lemma is identical , (2) whether the two are highly similar ($sim \geq 0.9$), (3) the longest common word/lemma sequence containing $T_i^{(1)}$ and $T_j^{(2)}$ such that at least one word in the sequence is a content word, (4) proportion of identical and highly similar ($sim \geq 0.9$) words in a neighborhood of 3 content words to the left and 3 content words to the right; we use two versions of this feature, one compares neighbors only in the same direction (i.e. left with left, right with right) and the other compares neighbors across the two directions, (5) similarly to dependency-based context, similarity in a max-weighted matching of all neighbors with $sim \in [0.5, 0.9)$ in the above $[-3, 3]$ window. For punctuation mark pairs, we use an additional feature indicating whether or not they both mark the end of their respective sentences.

4.2 Stage 2: Cooperation and Competition

We consider two groups of mutually cooperating words in a sentence: (1) words that belong to the same named entity, and (2) words in a sentence that are joined together to form a larger word in the other sentence (e.g., `state-of-the-art`). Speaking in terms of $T_i^{(1)}$, the goal is to be able to use any evidence present for a $(T_k^{(1)}, T_j^{(2)})$ alignment also as evidence for a $(T_i^{(1)}, T_j^{(2)})$ alignment if $T_i^{(1)}$ and $T_k^{(1)}$ both belong to such a group. We call $T_i^{(1)}$ and $T_k^{(1)}$ mutually cooperating words with respect to $T_j^{(2)}$ in such cases. Any word $T_l^{(1)} \in \boldsymbol{T}^{(1)}$ which is not a cooperating word for $T_i^{(1)}$ is a competing word: a word that can potentially make $(T_i^{(1)}, T_j^{(2)})$ a less viable alignment by having a larger stage 1 alignment probability in $(T_l^{(1)}, T_j^{(2)})$. We call a pair $(T_k^{(1)}, T_j^{(2)})$ a cooperating (competing) pair for $(T_i^{(1)}, T_j^{(2)})$ if $T_k^{(1)}$ is a cooperating (competing) word for $T_i^{(1)}$ with respect to $T_j^{(2)}$. With a reversal of word order and appropriate substitution of indexes, the above discussion

equally holds for $T_j^{(2)}$.

Given sets of stage 1 probabilities $\mathbf{\Phi}_{ij}^{cop}$ and $\mathbf{\Phi}_{ij}^{cmp}$ of cooperating and competing pairs for the pair $(T_i^{(1)}, T_j^{(2)})$, we employ three features to deal with scenario 2 of Section 2: (1) $\max(\phi_{ij}, \max(\mathbf{\Phi}_{ij}^{cop}))$: the greater of the pair's own stage 1 alignment probability and the highest among all cooperating pair probabilities, (2) $\max(\mathbf{\Phi}_{ij}^{cmp})$: the highest of all competing pair probabilities, and (3) a binary feature indicating which of the two above is larger.

To address scenario 3, we construct a weighted bipartite graph: nodes represent words in $\boldsymbol{T}^{(1)}$ and $\boldsymbol{T}^{(2)}$ and the weight of each edge represents the stage 1 alignment probability of a word pair in $\boldsymbol{T}^{(1)} \times \boldsymbol{T}^{(2)}$. We find a max-weighted bipartite matching $\boldsymbol{M}_\phi$ of word pairs in this graph. For each word pair, we employ a feature indicating whether or not it is in $\boldsymbol{M}_\phi$. The presence of $(T_i^{(1)}, T_j^{(2)})$ and $(T_k^{(1)}, T_l^{(2)})$ in $\boldsymbol{M}_\phi$, where all four words are similar, is a potential indicator that $(T_i^{(1)}, T_l^{(2)})$ and $(T_k^{(1)}, T_j^{(2)})$ are no longer viable alignments.

Low recall has traditionally been the primary weakness of supervised aligners (as we later show in Table 1). Our observation of the aligner's behavior on the *dev* set of the MSR alignment corpus (Brockett, 2007) suggests that this happens primarily due to highly similar word pairs being left unaligned even in the absence of competing pairs because of relatively low contextual evidence. Consequently, aligner performance suffers in sentences with few common or similar words. To promote high recall, we employ the higher of a word pair's own lexical similarity and the lexical similarity of the cooperating pair with the highest stage 1 probability as a stage 2 feature.

The stage 2 feature set is identical across word pair types, but as in stage 1, we train individual models for different pair types.

5 Experiments

5.1 System Evaluation

We report evaluation on two alignment data sets and extrinsic evaluation on two tasks: sentence similarity identification and paraphrase detection.

5.1.1 Alignment

We adopt the evaluation procedure for aligners reported in prior work (MacCartney et al., 2008; Thadani and McKeown, 2011; Yao et al., 2013a).

	Aligner	P %	R %	F_1 %	E %
MSR	MacCartney et al. (2008)	85.4	85.3	85.3	21.3
	Thadani & McKeown (2011)	89.5	86.2	87.8	33.0
	Yao et al. (2013a)	93.7	84.0	88.6	35.3
	Yao et al. (2013b)	92.1	82.8	86.8	29.1
	Sultan et al. (2014a)	93.7	**89.8**	91.7	43.8
	Our Aligner	**95.4**	89.0	**92.1**	**47.3**
EDB++	Thadani et al. (2012)	76.6	83.8	79.2	12.2
	Yao et al. (2013a)	91.3	82.0	86.4	15.0
	Yao et al. (2013b)	90.4	81.9	85.9	13.7
	Sultan et al. (2014a)	**93.5**	82.5	87.6	**18.3**
	Our Aligner	92.1	**85.2**	**88.5**	**18.3**

Table 1: Performance on two alignment data sets. Improvements in F_1 are statistically significant.

Data. The MSR alignment corpus (Brockett, 2007) contains 800 *dev* and 800 *test* sentence pairs from the PASCAL RTE 2006 challenge. Each pair is aligned by three human annotators; Fleiss Kappa agreement of about 0.73 ("substantial agreement") is reported on both sets. Following prior work, we only consider the *sure* alignments, take the majority opinion on each word pair, and leave out three-way disagreements.

The Edinburgh++ corpus (Thadani et al., 2012) contains 714 training and 306 test sentence pairs. Each test pair is aligned by two annotators and the final gold alignments consist of a random but even selection of the two sets of annotations.

Evaluation metrics. Our primary evaluation metrics are macro-averaged precision (*P*), recall (*R*) and F_1 score. A fourth metric E measures the proportion of sentence pairs for which the system alignments are identical to the gold alignments.

Model setup. For each corpus, we train our model using the dev set and evaluate on the test set. We use the logistic regression implementation of Scikit-learn (Pedregosa et al., 2011) and use leave-one-out cross-validation on the dev pairs to set the regularization parameter C.

Results. Table 1 shows the performance of different aligners on the two test sets. Our aligner demonstrates the best overall performance in terms of both F_1 and E. Wilcoxon signed-rank tests (with Pratt's treatment for zero-difference pairs) show that the improvements in F_1 over the previous best aligner (Sultan et al., 2014a) are statistically significant at $p < 0.01$ for both test sets.

5.1.2 Identification of Sentence Similarity

Given two input sentences, the goal in this task, known also as Semantic Textual Similarity (STS), is to output a real-valued semantic similarity score.

System	Pearson's r	Rank
Han et al. (2013)	**73.7**	1
Yao et al. (2013a)	46.2	66
Sultan et al. (2014a)	67.2	7
Our Aligner	67.8	4

Table 2: STS results. Performances of past systems are reported by Sultan et al. (2014a).

Data. To be able to directly compare with past aligners, we select three data sets (*headlines*: pairs of news headlines; *OnWN*, *FNWN*: gloss pairs) from the 2013 *SEM STS corpus (Agirre et al., 2013), containing 1500 sentence pairs in total. Sultan et al. (2014a) reports the performance of two state-of-the-art aligners on these pairs.

Evaluation metric. At SemEval, STS systems output a similarity score in $[0, 5]$. For each individual test set, the Pearson product-moment correlation coefficient (Pearson's r) is computed between system scores and human annotations. The final evaluation metric is a weighted sum of r's over all test sets, where the weight assigned to a set is proportional to its number of pairs.

Method. Being a logistic regression model, stage 2 of our aligner assigns each word pair an alignment probability. For STS, we compute a length-normalized sum of alignment probabilities of content word pairs across the two sentences. We include all pairs with probability > 0.5; the remaining pairs are included in decreasing order of their probabilities and already included words are ignored. Following (Sultan et al., 2014a), we normalize by dividing with the harmonic mean of the numbers of content words in the two sentences.

Results. Table 2 shows the performance of different aligners on the three STS 2013 test sets. We also show the performance of the contest-winning system (Han et al., 2013). Our STS system demonstrates a weighted correlation of 67.8%, which is better than similar STS systems based on the two previous best aligners. The difference with the next best aligner is statistically significant at $p < 0.05$ (two-sample one-tailed z-test). Overall, our system outperforms 86 of the 89 participating systems.

5.1.3 Paraphrase Detection

Given two input sentences, the goal in this task is to determine if their meanings are the same.

Data. The MSR paraphrase corpus (Dolan et al., 2004) contains 4076 *dev* and 1725 *test* sentence pairs; a paraphrase label (*true*/*false*) for each pair is provided by human annotators.

System	A %	P %	R %	F_1 %
Madnani et al. (2012)	**77.4**	**79.0**	89.9	**84.1**
Yao et al. (2013a)	70.0	72.6	88.1	79.6
Yao et al. (2013b)	68.1	68.6	**95.8**	79.9
Sultan et al. (2014a)	73.4	76.6	86.4	81.2
Our Aligner	73.2	75.3	88.8	81.5

Table 3: Paraphrase results. Performances of past systems are taken from (Sultan et al., 2014a).

Evaluation Metrics. We report performance in terms of: (1) accuracy in classifying the sentences into *true* and *false* classes (A), and (2) true class precision (P), recall (R) and F_1 score.

Method. Following prior aligners (MacCartney et al., 2008; Yao et al., 2013b; Sultan et al., 2014a), we output a *true* decision for a test sentence pair iff the length-normalized alignment score for the pair exceeds a threshold derived from the dev set.

Results. The top row of Table 3 shows the best result by any system on the MSR test set. Among all aligners (all other rows), ours achieves the best F_1 score and the second best accuracy.

We report paraphrase detection results primarily to allow comparison with past aligners. However, this simplistic application to a complex task only gives a ballpark estimate of an aligner's quality.

	Model	P %	R %	F_1 %	E %
MSR	Two-Stage Model	**95.4**	**89.0**	**92.1**	**47.3**
	Stage 1 Only	92.9	85.6	89.1	28.0
EDB++	Two-Stage Model	92.1	**85.2**	**88.5**	**18.3**
	Stage 1 Only	**93.0**	79.0	85.4	13.7

Table 4: Performance with and without stage 2.

5.2 Ablation

We perform ablation tests to find out how important (1) the two-stage framework, and (2) the different features are for our aligner.

5.2.1 Results without Stage 2

Stage 1 of our aligner can operate as an aligner by itself by mapping each alignment probability to a 0/1 alignment decision based on a threshold of 0.5. From a design perspective, this is an aligner that does not address scenarios 2 and 3 of Section 2.

The performance of the aligner with and without stage 2 is shown in Table 4. On each test set, the F_1 and E scores increase with the addition of stage 2. On the MSR test set, performance improves along all dimensions. On the Edinburgh++ test set, the

	MSR			EDB++		
Features	P %	R %	F_1 %	P %	R %	F_1 %
All Features	95.4	**89.0**	**92.1**	92.1	**85.2**	**88.5**
- Lexical	95.1	82.8	88.5	90.9	84.0	87.3
- Resources	**96.0**	87.0	91.3	**92.2**	84.3	88.1
- Contextual	89.0	79.2	83.9	89.9	66.3	76.3
- Dependency	95.3	88.2	91.6	91.9	84.9	88.3
- Surface	94.4	85.6	89.8	90.6	76.9	83.2
- Word-Based	94.6	87.7	91.0	92.0	85.1	88.4
- Entity-Based	95.5	**89.0**	**92.1**	92.1	85.1	**88.5**

Table 5: Results without different stage 1 features.

	MSR			EDB++		
Features	P %	R %	F_1 %	P %	R %	F_1 %
All Features	95.4	89.0	**92.1**	92.1	85.2	**88.5**
- ϕ values	87.3	**90.7**	88.9	86.4	**86.9**	86.7
- Matching	95.3	87.9	91.5	92.3	84.6	88.3
- Word Sim	**95.5**	88.4	91.8	**92.7**	84.7	**88.5**

Table 6: Results without different stage 2 features.

precision drops a little, but this effect is offset by a larger improvement in recall. These results show that stage 2 is central to the aligner's success.

5.2.2 Without Different Stage 1 Features

We exclude different stage 1 features (which fall into one of two groups: lexical and contextual) and examine the resulting model's performance. Table 5 shows the results. The subtraction sign represents the exclusion of the corresponding feature.

Without any lexical feature (i.e., if the model relies only on contextual features), both precision and recall decrease, resulting in a considerable overall performance drop. Exclusion of word similarity resources (i.e. embeddings and PPDB) improves precision, but again harms overall performance.

Without any contextual features, the model suffers badly in both precision and recall. The extreme overall performance degradation indicates that contextual features are more important for the aligner than lexical features. Leaving out surface-form neighbors results in a larger performance drop than when dependency-based neighbors are excluded, pointing to a more robust role of the former group in representing context. Finally, our entity-based representation of context neither helps nor harms system performance, but relying only on entity-based neighbors has detrimental effects. Factoring in semantic similarities of named entities should improve the utility of these features.

5.2.3 Without Different Stage 2 Features

Table 6 shows the aligner's performance after the exclusion of different stage 2 features. Leaving out the stage 1 alignment probabilities harms overall performance the most by causing a large drop in precision. Exclusion of the maximum-weighted bipartite matching feature results in worse recall and overall performance. The lexical similarity feature improves overall results only on the MSR test set but increases recall on both test sets.

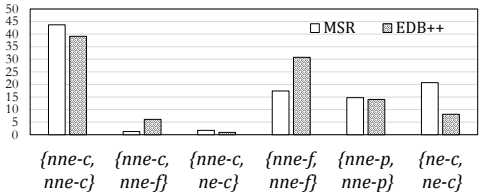

Figure 4: % distribution of aligned word pair types; *nne*: non-named entity, *ne*: named entity, *c*: content word, *f*: function word, *p*: punctuation mark.

	MSR			EDB++		
Pair Type	P %	R %	F_1 %	P %	R %	F_1 %
{*nne-c*, *nne-c*}	95.7	84.3	89.7	92.2	89.2	90.7
{*nne-c*, *nne-f*}	100.0	2.7	5.3	61.4	7.7	13.6
{*nne-c*, *ne-c*}	89.2	66.7	76.3	71.9	43.4	54.1
{*nne-f*, *nne-f*}	90.7	86.0	88.3	93.4	86.5	89.8
{*nne-p*, *nne-p*}	99.4	99.2	99.3	93.0	91.5	92.2
{*ne-c*, *ne-c*}	96.2	97.8	97.0	90.9	94.2	92.6

Table 7: Performance on different word pair types.

5.3 Error Analysis

We examine the aligner's performance on different word pair types. Figure 4 shows the % distribution of word pair types with at least 20 aligned instances in at least one test set. These six types account for more than 99% of all alignments in both test sets.

Table 7 shows the results. We ignore punctuation mark pairs in the following discussion. Performance is worst on the two rarest types: {*nne-c*, *nne-f*} and {*nne-c*, *ne-c*}, due primarily to very low recall. A relatively low availability of positive examples in the training sets (in the hundreds, in contrast to thousands of examples for each of the other three types) is a primary factor affecting classifier performance on these two pair types. The {*nne-c*, *nne-f*} pairs, nonetheless, are intrinsically the most difficult type for a word aligner because they occur frequently as part of phrasal alignments. On {*nne-c*, *ne-c*} pairs, errors also occur due to failure in recognition of certain named entity types (e.g., acronyms and multiword named entities) and the aligner's lack of world knowledge (e.g., in `daughter` ↔ `Chelsea`).

Low recall remains the primary issue, albeit to a much lesser extent, for {*nne-c*, *nne-c*} and {*nne-f*, *nne-f*} pairs. For the former, two major sources of error are: (1) inability to utilize contextual evidence outside the local neighborhood examined by the aligner, and (2) failure to address one-to-many alignments. Low recall for the latter follows naturally, as function word alignment is heavily dependent on related content word alignment. On {*ne-c*, *ne-c*} pairs the aligner performs the best, but still suffers from the two above issues.

6 Related Work

We mentioned major standalone monolingual aligners and briefly discussed their working principles in Section 2. There are, however, at least two additional groups of related work which can inform future research on monolingual alignment. First, alignment is often performed in the context of extrinsic tasks, e.g., textual entailment recognition (Wang and Manning, 2010), question answering (Heilman and Smith, 2010), discourse generation (Roth and Frank, 2012) and redundancy detection (Thadani and McKeown, 2008). Such systems may contain useful design elements yet to be utilized by standalone aligners. Second, a large body of work exists in the bilingual alignment literature (Och and Ney, 2003; Blunsom and Cohn, 2006; Chang et al., 2014), elements of which (such as the machine learning models) can be useful for monolingual aligners (see (Yao et al., 2013a) for an example).

7 Conclusions and Future Work

We present a two-stage classification framework for monolingual alignment that demonstrates top results in intrinsic and extrinsic evaluation experiments. While our work focuses primarily on word alignment, given a mechanism to compute phrasal similarity, the notion of cooperating words can be exploited to extend our model for phrasal alignment. Another important future direction is the construction of a robust representation of context, as our model currently utilizes contextual information only within a local neighborhood of a predefined size and therefore fails to utilize long-distance semantic relationships between words. Incorporating a single background model which is trained on all word pair types might also improve performance, especially on types that are rare in the training data. Finally, studying the explicit requirements of different extrinsic tasks can shed light on the design of a robust aligner.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Numbers EHR/0835393 and EHR/0835381.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 Shared Task: Semantic Textual Similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, *SEM '13, pages 32-43, Atlanta, Georgia, USA.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 252-263, Denver, Colorado, USA.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't Count, Predict! A Systematic Comparison of Context-Counting vs. Context-Predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 238-247, Baltimore, Maryland, USA.

Phil Blunsom and Trevor Cohn. 2006. Discriminative Word Alignment with Conditional Random Fields. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 65-72, Sydney, Australia.

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

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, vol. 49, pages 1-47.

Yin-Wen Chang, Alexander M. Rush, John DeNero, and Michael Collins. 2014. A Constrained Viterbi Relaxation for Bidirectional Word Alignment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 14811490, Baltimore, Maryland, USA.

Ido Dagan and Oren Glickman. 2004. Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. In *Proceedings of the PASCAL Workshop on Learning Methods for Text Understanding and Mining*, Grenoble, France.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase Identification as Probabilistic Quasi-Synchronous Recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 468-476, Singapore.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 449-454, Genoa, Italy.

Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Cer, Daniel Ramage, Chlo Kiddon, and Christopher D. Manning. 2007. Aligning Semantic Graphs for Textual Inference and Machine Reading. In *Proceedings of the AAAI Spring Symposium*, pages 468-476, Stanford, California, USA.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of the International Conference on Computational Linguistics*, pages 350-356, Geneva, Switzerland.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 758-764, Atlanta, Georgia, USA.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-Scale Learning of Word Relatedness with Constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1406-1414, Beijing, China.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, *SEM '13, pages 44-52, Atlanta, Georgia, USA.

Michael Heilman and Noah A. Smith. 2010. Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 10111019, Los Angeles, California, USA.

Andrew Hickl and Jeremy Bensley. 2007. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171-176, Prague, Czech Republic.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning Knowledge Graphs for Question Answering through Conversational Dialog. In *Proceedings of the the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, USA.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A Phrase-Based Alignment Model for Natural Language Inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802-811, Honolulu, Hawaii, USA.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining Machine Translation Metrics for Paraphrase Identification. In *Proceedings of 2012 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 182-190, Montreal, Canada.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations Workshop*, Scottsdale, Arizona, USA.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to Grade Short Answer Questions Using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 752-762, Portland, Oregon, USA.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):1951, MIT Press.

Sebastian Padó, Michel Galley, Dan Jurafsky, and Chris Manning. 2009. Robust Machine Translation Evaluation with Entailment Features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297-305, Singapore.

Sebastian Padó, Tae-Gil Noh, Asher Stern, Rui Wang, and Roberto Zanoli. 2015. Design and Realization of a Modular Architecture for Textual Entailment. *Natural Language Engineering*, 21 (2), pages 167-200, Cambridge University Press.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, pages 2825-2830.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 337-346, Hyderabad, India.

Michael Roth and Anette Frank. 2012. Aligning Predicates across Monolingual Comparable Texts using Graph-based Clustering. In *Proceedings of the 20th International Conference on World Wide Web*, pages pages 171182, Jeju Island, Korea.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014a. Back to Basics for Monolingual Alignment: Exploiting Word Similarity and Contextual Evidence. *Transactions of the Association for Computational Linguistics*, 2 (May), pages 219-230.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014b. DLS@CU: Sentence Similarity from Word Alignment. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 241-246, Dublin, Ireland.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 148-153, Denver, Colorado, USA.

Kapil Thadani and Kathleen McKeown. 2008. A Framework for Identifying Textual Redundancy. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 873880, Manchester, UK.

Kapil Thadani and Kathleen McKeown. 2011. Optimal and Syntactically-Informed Decoding for Monolingual Phrase-Based Alignment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 254-259, Portland, Oregon, USA.

Kapil Thadani, Scott Martin, and Michael White. 2012. A Joint Phrasal and Dependency Model for Paraphrase Alignment. In *Proceedings of COLING 2012*, pages 1229-1238, Mumbai, India.

Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 11641172, Beijing, China.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013a. A Lightweight and High Performance Monolingual Word Aligner. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 702-707, Sofia, Bulgaria.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013b. Semi-Markov Phrase-based Monolingual Alignment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 590-600, Seattle, Washington, USA.