Exploiting Sentence Similarities for Better Alignments

Tao Li and **Vivek Srikumar**
University of Utah
{tli,svivek}@cs.utah.edu

Abstract

We study the problem of jointly aligning sentence constituents and predicting their similarities. While extensive sentence similarity data exists, manually generating reference alignments and labeling the similarities of the aligned chunks is comparatively onerous. This prompts the natural question of whether we can exploit easy-to-create sentence level data to train better aligners. In this paper, we present a model that learns to jointly align constituents of two sentences and also predict their similarities. By taking advantage of both sentence and constituent level data, we show that our model achieves state-of-the-art performance at predicting alignments and constituent similarities.

1 Introduction

The problem of discovering semantic relationships between two sentences has given birth to several NLP tasks over the years. Textual entailment (Dagan et al., 2013, *inter alia*) asks about the truth of a hypothesis sentence given another sentence (or more generally a paragraph). Paraphrase identification (Dolan et al., 2004, *inter alia*) asks whether two sentences have the same meaning. Foregoing the binary entailment and paraphrase decisions, the semantic textual similarity (STS) task (Agirre et al., 2012) asks for a numeric measure of semantic equivalence between two sentences. All three tasks have attracted much interest in the form of shared tasks.

While various approaches have been proposed to predict these sentence relationships, a commonly employed strategy (Das and Smith, 2009; Chang et al., 2010a) is to postulate an alignment between constituents of the sentences and use this alignment to make the final prediction (a binary decision or a numeric similarity score). The implicit assumption in such approaches is that better constituent alignments can lead to better identification of semantic relationships between sentences.

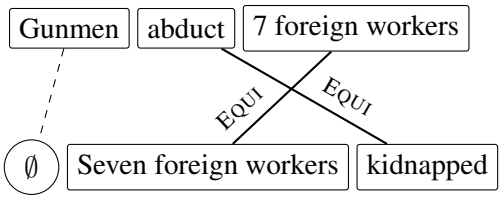

Figure 1: Example constituent alignment. The solid lines represent aligned constituents (here, both labeled equivalent). The chunk *Gunmen* is unaligned.

Constituent alignments serve two purposes. First, they act as an intermediate representation for predicting the final output. Second, the alignments help interpret (and debug) decisions made by the overall system. For example, the alignment between the sentences in Figure 1 can not only be useful to determine the equivalence of the two sentences, but also help reason about the predictions.

The importance of this intermediate representation led to the creation of the *interpretable* semantic textual similarity task (Agirre et al., 2015a) that focuses on predicting chunk-level alignments and similarities. However, while extensive resources exist for sentence-level relationships, human annotated chunk-aligned data is comparatively smaller.

In this paper, we address the following question: can we use sentence-level resources to better pre-

dict constituent alignments and similarities? To answer this question, we focus on the semantic textual similarity (STS) task and its interpretable variant. We propose a joint model that aligns constituents and integrates the information across the aligned edges to predict both constituent and sentence level similarity. The key advantage of modeling these two problems jointly is that, during training, the sentence-level information can provide feedback to the constituent-level predictions.

We evaluate our model on the SemEval-2016 task of interpretable STS. We show that even without the sentence information, our joint model that uses constituent alignments and similarities forms a strong baseline. Further, our easily extensible joint model can incorporate sentence-level similarity judgments to produce alignments and chunk similarities that are comparable to the best results in the shared task.

In summary, the contributions of this paper are:

1. We present the first joint model for predicting constituent alignments and similarities. Our model can naturally take advantage of the much larger sentence-level annotations.

2. We evaluate our model on the SemEval-2016 task of interpretable semantic similarity and show state-of-the-art results.

2 Problem Definition

In this section, we will introduce the notation used in the paper using the sentences in Figure 1 as a running example. The input to the problem is a pair of sentences, denoted by $\mathbf{x}$. We will assume that the sentences are chunked (Tjong Kim Sang and Buchholz, 2000) into constituents. We denote the chunks using subscripts. Thus, the input $\mathbf{x}$ consists of two sequences of chunks $\mathbf{s} = (s_1, s_2, \cdots)$ and $\mathbf{t} = (t_1, t_2, \cdots)$ respectively. In our running example, we have $\mathbf{s} =$ (*Gunmen*, *abduct*, *seven foreign workers*) and $\mathbf{t} =$ (*Seven foreign workers*, *kidnapped*).

The output consists of three components:

1. **Alignment**: The alignment between a pair of chunks is a labeled, undirected edge that explains the relation that exists between them. The labels can be one of EQUI (semantically equivalent), OPPO (opposite meaning in context), SPE1, SPE2 (the chunk from $\mathbf{s}$ is more specific than the one from $\mathbf{t}$ and vice versa), SIMI (similar meaning, but none of the previous ones) or REL (related, but none of the above)[1]. In Figure 1, we see two EQUI edges. A chunk from either sentence can be unaligned, as in the case of the chunk *Gunmen*.

   We will use $\mathbf{y}$ to denote the alignment for an input $\mathbf{x}$. The alignment $\mathbf{y}$ consists of a sequence of triples of the form (s_i, t_j, l). Here, s_i and t_j denote a pair of chunks that are aligned with a label l. For brevity, we will include unaligned chunks into this format using a special null chunk and label to indicate that a chunk is unaligned. Thus, the alignment for our running example contain the triple $(\textit{Gunmen}, \emptyset, \emptyset)$.

2. **Chunk similarity**: Every aligned chunk is associated with a relatedness score between zero and five, denoting the range from unrelated to equivalent. Note that even chunks labeled OPPO can be assigned a high score because the polarity is captured by the label rather than the score. We will denote the chunk similarities using $\mathbf{z}$, comprising of numeric $z_{i,j,l}$ for elements of the corresponding alignment $\mathbf{y}$. For an unaligned chunk, the corresponding similarity z is fixed to zero.

3. **Sentence similarity**: The pair of sentences is associated with a scalar score from zero to five, to be interpreted as above. We will use r to denote the sentence similarity for an input $\mathbf{x}$.

Thus, the prediction problem is the following: Given a pair of chunked sentences $\mathbf{x} = (\mathbf{s}, \mathbf{t})$, predict the alignment $\mathbf{y}$, the alignment similarities $\mathbf{z}$ and the sentence similarity r. Note that this problem definition integrates the canonical semantic textual similarity task (only predicting r) and its interpretable variant (predicting both $\mathbf{y}$ and $\mathbf{z}$) into a single task.

[1] We refer the reader to the guidelines of the task (Agirre et al., 2015a) for further details on these labels. Also, for simplicity, in this paper, we ignore the factuality and polarity tags from the interpretable task.

3 Predicting Alignments and Similarities

This section describes our model for predicting alignments, alignment scores, and the sentence similarity scores for a given pair of sentences. We will assume that learning is complete and we have all the scoring functions we need and defer discussing the parameterization and learning to Section 4.

We frame the problem of inference as an instance of an integer linear program (ILP). We will first see the scoring functions and the ILP formulation in Section 3.1. Then, in Section 3.2, we will see how we can directly read off the similarity scores at both chunk and sentence level from the alignment.

3.1 Alignment via Integer Linear Programs

We have two kinds of 0-1 inference variables to represent labeled aligned chunks and unaligned chunks.

We will use the inference variables $\mathbb{1}_{i,j,l}$ to denote the decision that chunks s_i and t_j are aligned with a label l. To allow chunks to be unaligned, the variables $\mathbb{1}_{i,0}$ and $\mathbb{1}_{0,j}$ denote the decisions that s_i and t_j are unaligned respectively.

Every inference decision is scored by the trained model. Thus, we have $\text{score}(i,j,l)$, $\text{score}(i,0)$ and $\text{score}(0,j)$ for the three kinds of inference variables respectively. All scores are of the form $A\left(\mathbf{w}^T\Phi\left(\cdot,\mathbf{s},\mathbf{t}\right)\right)$, where $\mathbf{w}$ is a weight vector that is learned, $\Phi\left(\cdot,\mathbf{s},\mathbf{t}\right)$ is a feature function whose arguments include the constituents and labels in question, and A is a sigmoidal activation function that flattens the scores to the range $[0,5]$. In all our experiments, we used the function $A(x)=\frac{5}{1+e^{-x}}$.

The goal of inference is to find the assignment to the inference variables that maximizes total score. That is, we seek to solve

$$\begin{aligned}\arg\max_{\mathbb{1}\in\mathcal{C}} \quad & \sum_{i,j,l}\text{score}(i,j,l)\mathbb{1}_{i,j,l} \\ & +\sum_{i}\text{score}(i,0)\mathbb{1}_{i,0} \\ & +\sum_{j}\text{score}(0,j)\mathbb{1}_{0,j} \qquad (1)\end{aligned}$$

Here $\mathbb{1}$ represents all the inference variables together and $\mathcal{C}$ denotes the set of all valid assignments to the variables, defined by the following set of constraints:

1. A pair of chunks can have *at most* one label.
2. Either a chunk can be unaligned or it should participate in a labeled alignment with exactly one chunk of the other sentence.

We can convert these constraints into linear inequalities over the inference variables using standard techniques for ILP inference (Roth and Yih, 2004)[2]. Note that, by construction, there is a one-to-one mapping from an assignment to the inference variables $\mathbb{1}$ and the alignment $\mathbf{y}$. In the rest of the paper, we use these two symbols interchangeably, using $\mathbb{1}$ referring details of inference and $\mathbf{y}$ referring to the alignment as a sequence of labeled edges.

3.2 From Alignments to Similarities

To complete the prediction, we need to compute the numeric chunk and sentence similarities given the alignment $\mathbf{y}$. In each case, we make modeling assumptions about how the alignments and similarities are related, as described below.

Chunk similarities To predict the chunk similarities, we assume that the label-specific chunk similarities of aligned chunks *are* the best edge-weights for the corresponding inference variables. That is, for a pair of chunks (s_i,t_j) that are aligned with a label l, the chunk pair similarity $z_{i,j,l}$ is the coefficient associated with the corresponding inference variable. If the alignment edge indicates an unaligned chunk, then the corresponding score is zero. That is,

$$z_{i,j,l}=\begin{cases}A\left(\mathbf{w}^T\Phi\left(s_i,t_j,l,\mathbf{s},\mathbf{t}\right)\right) & \text{if } l\neq\emptyset \\ 0 & \text{if } l=\emptyset.\end{cases} \qquad (2)$$

But can chunk similarities directly be used to find good alignments? To validate this assumption, we performed a pilot experiment on the chunk aligned part of our training dataset. We used the gold standard chunk similarities as scores of the inference variables in the integer program in Eq. 1, with the variables associated with unaligned chunks being scored zero. We found that this experiment gives a near-perfect typed alignment F-score of 0.9875.

[2]While it may be possible to find the score maximizing alignment in the presence of these constraints using dynamic programming (say, a variant of the Kuhn-Munkres algorithm), we model inference as an ILP to allow us the flexibility to explore more sophisticated output interactions in the future.

The slight disparity is because the inference only allows 1-to-1 matches between chunks (constraint 2), which does not hold in a small number of examples.

Sentence similarities Given the aligned chunks $\mathbf{y}$, the similarity between the sentences $\mathbf{s}$ and $\mathbf{t}$ (*i.e.*, in our notation, r) is the weighted average of the chunk similarities (*i.e.*, $z_{i,j,l}$). Formally,

$$r = \frac{1}{|\mathbf{y}|} \sum_{(s_i,t_j,l)\in\mathbf{y}} \alpha_l z_{i,j,l}. \quad (3)$$

Note that the weights α_l depend only on the labels associated with the alignment edge and are designed to capture the polarity and strength of the label. Eq. 3 bridges sentence similarities and chunk similarities. During learning, this provides the feedback from sentence similarities to chunk similarities. The values of the α's can be learned or fixed before learning commences. To simplify our model, we choose the latter approach . Section 5 gives more details.

Features To complete the description of the model, we now describe the features that define the scoring functions. We use standard features from the STS literature (Karumuri et al., 2015; Agirre et al., 2015b; Banjade et al., 2015).

For a pair of chunks, we extract the following similarity features: (1) Absolute cosine similarities of GloVe embeddings (Pennington et al., 2014) of head words, (2) WordNet based Resnik (Resnik, 1995), Leacock (Leacock and Chodorow, 1998) and Lin (Lin, 1998) similarities of head words, (3) Jaccard similarity of content words and lemmas. In addition, we also add indicators for: (1) the part of speech tags of the pair of head words, (2) the pair of head words being present in the lexical large section of the Paraphrase Database (Ganitkevitch et al., 2013), (3) a chunk being longer than the other while both are not named entity chunks, (4) a chunk having more content words than the other, (5) contents of one chunk being a part of the other, (6) having the same named entity type or numeric words, (7) sharing synonyms or antonyms, (8) sharing conjunctions or prepositions, (9) the existence of unigram/bigram/trigram overlap, (10) if only one chunk has a negation, and (11) a chunk having extra content words that are also present in the other sentence.

For a chunk being unaligned, we conjoin an indicator that the chunk is unaligned with the part of speech tag of its head word.

3.3 Discussion

In the model proposed above, by predicting the alignment, we will be able to deterministically calculate both chunk and sentence level similarities. This is in contrast to other approaches for the STS task, which first align constituents and then extract features from alignments to predict similarities in a pipelined fashion. The joint prediction of alignment and similarities allows us to address the primary motivation of the paper, namely using the abundant sentence level data to train the aligner and scorer.

The crucial assumption that drives the joint model is that the *same* set of parameters that can discover a good alignment can also predict similarities. This assumption – similar to the one made by Chang et al. (2010b) – and the associated model described above, imply that the goal of learning is to find parameters that drive the inference towards good alignments and similarities.

4 Learning the Alignment Model

Under the proposed model, the alignment directly predicts the chunk and sentence similarities as well. We utilize two datasets to learn the model:

1. The **alignment dataset** D_A consists of fully annotated aligned chunks and respective chunk similarity scores.

2. The **sentence dataset** D_S that consists of pairs of sentences where each pair is labeled with a numeric similarity score between zero and five.

The goal of learning is to use these two datasets to train the model parameters. Note that unlike standard multi-task learning problems, the two tasks in our case are tightly coupled both in terms of their definition and via the model described in Section 3.

We define three types of loss functions corresponding to the three components of the final output (*i.e.*, alignment, chunk similarity and sentence similarity). Naturally, for each kind of loss, we assume that we have the corresponding ground truth. We will denote ground truth similarity scores and alignments using asterisks. Also, the loss functions

defined below depend on the weight vector $\mathbf{w}$, but this is not shown to simplify notation.

1. The **alignment loss** L_a is a structured loss function that penalizes alignments that are far away from the ground truth. We used the structured hinge loss (Taskar et al., 2004; Tsochantaridis et al., 2005) for this purpose.

$$L_a(\mathbf{s}, \mathbf{t}, \mathbf{y}^*) = \max_{\mathbf{y}} \mathbf{w}^T \Phi(\mathbf{s}, \mathbf{t}, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^*) - \mathbf{w}^T \Phi(\mathbf{s}, \mathbf{t}, \mathbf{y}^*).$$

   Here, Δ refers to the Hamming distance between the alignments.

2. The **chunk score loss** L_c is designed to penalize errors in predicted chunk level similarities. To account for cases where chunk boundaries may be incorrect, we define this loss as the sum of squared errors of token similarities. However, neither our output nor the gold standard similarities are at the granularity of tokens. Thus, to compute the loss, we project the chunk scores $z_{i,j,l}$ for an aligned chunk pair (s_i, t_j, l) to the tokens that constitute the chunks by equally partitioning the scores among all possible internal alignments. In other words, for a token w_i in the chunk s_i and token w_j in chunk s_j, we define token similarity scores as

$$z(w_i, w_j, l) = \frac{z_{i,j,l}}{N(s_i, t_j)}$$

   Here, the normalizing function N is the product of the number of tokens in the chunks[3]. Note that this definition of the token similarity scores applies to both predicted and gold standard similarities. Unaligned tokens are associated with a zero score.

   We can now define the loss for a token pair $(w_i, w_j) \in (\mathbf{s}, \mathbf{t})$ and a label l as the squared error of their token similarity scores:

$$l(w_i, w_j, l) = (z(w_i, w_j, l) - z^*(w_i, w_j, l))^2$$

   The chunk loss score L_c for a sentence pair is the sum of all the losses over all pairs of tokens and labels.

$$L_c(\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{y}^*, \mathbf{z}, \mathbf{z}^*) = \sum_{w_i, w_j, l} l(w_i, w_j, l)$$

3. The **sentence similarity loss** L_s provides feedback to the aligner by penalizing alignments that are far away from the ground truth in their similarity assessments. For a pair of sentences $(\mathbf{s}, \mathbf{t})$, given the ground truth sentence similarity r^* and the predicted sentence similarity r (using Equation (3)), the sentence similarity loss is the squared error:

$$L_s(\mathbf{s}, \mathbf{t}, r^*) = (r - r^*)^2.$$

Our learning objective is the weighted combination of the above three components and a ℓ_2 regularizer on the weight vector. The importance of each type of loss is controlled by a corresponding hyperparameter: λ_a, λ_c and λ_s respectively.

Learning algorithm We have two scenarios to consider: with only alignment dataset D_A, and with both D_A and sentence dataset D_S. Note that even if we train only on the alignment dataset D_A, our learning objective is not convex because the activation function is sigmoidal (in Section 3.1).

In both cases, we use stochastic gradient descent with minibatch updates as the optimizer. In the first scenario, we simply perform the optimization using the alignment and the chunk score losses. We found by preliminary experiments on training data that initializing the weights to one performed best.

Algorithm 1 Learning alignments and similarities, given alignment dataset D_A and sentence similarity dataset D_S. See the text for more details.

1: Initialize all weights to one.
2: $\mathbf{w}^0 \leftarrow SGD(D_A)$: Train an initial model
3: Use $\mathbf{w}^0$ to predict alignments on examples in D_S. Call this $\widehat{D_S}$.
4: $\mathbf{w} \leftarrow SGD(D_A \cup \widehat{D_S})$: Train on both sets of examples.
5: **return** $\mathbf{w}$

When we have both D_A and D_S (Algorithm 1), we first initialize the model on the alignment data

[3]Following the official evaluation of the interpretable STS task, we also experimented with the $\max(|s_i|, |t_j|)$ for the normalizer, but we found via cross validation that the product performs better.

only. Using this initial model, we hypothesize alignments on all examples in D_S to get fully labeled examples. Then, we optimize the full objective (all three loss terms) on the combined dataset. Because our goal is to study the impact on the chunk level predictions, in the full model, the sentence loss does not play a part on examples from D_A.

5 Experiments and Results

The primary research question we seek to answer via experiments is: Can we better predict chunk alignments and similarities by taking advantage of sentence level similarity data?

Datasets We used the training and test data from the 2016 SemEval shared tasks of predicting semantic textual similarity (Agirre et al., 2016a) and interpretable STS (Agirre et al., 2016b), that is, tasks 1 and 2 respectively. For our experiments, we used the headlines and images sections of the data. The data for the interpretable STS task, consisting of manually aligned and scored chunks, provides the alignment datasets for training (D_A). The headlines section of the training data consists for 756 sentence pairs, while the images section consists for 750 sentence pairs. The data for the STS task acts as our sentence level training dataset (D_S). For the headlines section, we used the 2013 headlines test set consisting of 750 sentence pairs with gold sentence similarity scores. For the images section, we used the 2014 images test set consisting of 750 examples. We evaluated our models on the official Task 2 test set, consisting of 375 sentence pairs for both the headlines and images sections. In all experiments, we used gold standard chunk boundaries if they are available (*i.e.*, for D_A).

Pre-processing We pre-processed the sentences with parts of speech using the Stanford CoreNLP toolkit (Manning et al., 2014). Since our setting assumes that we have the chunks as input, we used the Illinois shallow parser (Clarke et al., 2012) to extract chunks from D_S. We post-processed the predicted chunks to correct for errors using the following steps: 1. Split on punctuation; 2. Split on verbs in NP; 3. Split on nouns in VP; 4. Merge PP+NP into PP; 5. Merge VP+PRT into VP if the PRT chunk is not a preposition or a subordinating conjunction; 6. Merge SBAR+NP into SBAR; and 7. Create new contiguous chunks using tokens that are marked as being outside a chunk by the shallow parser. We found that using the above post-processing rules, improved the F1 of chunk accuracy from 0.7865 to 0.8130. We also found via cross-validation that this post-processing improved overall alignment accuracy. The reader may refer to other STS resources (Karumuri et al., 2015) for further improvements along this direction.

Experimental setup We performed stochastic gradient descent for 200 epochs in our experiments, with a mini-batch size of 20. We determined the three λ's using cross-validation, with different hyperparameters for examples from D_A and D_S. Table 1 lists the best hyperparameter values. For performing inference, we used the Gurobi optimizer[4].

Setting	$\lambda_a, \lambda_c, \lambda_s$
headlines, D_A	100, 0.01, N/A
headlines, D_S	0.5, 1, 50
images, D_A	100, 0.01, N/A
images, D_S	5, 2.5, 50

Table 1: Hyperparameters for the various settings, chosen by cross-validation. The alignment dataset do not have a λ associated with the sentence loss.

As noted in Section 3.1, the parameter α_l combines chunk scores into sentence scores. To find these hyper-parameters, we used a set of 426 sentences from the from the headlines training data that had both sentence and chunk annotation. We simplified the search by assuming that α_{Equi} is always 1.0 and all labels other than OPPO have the same α. Using grid search over $[-1, 1]$ in increments of 0.1, we selected α's that gave us the highest Pearson correlation for sentence level similarities. The best α's (with a Pearson correlation of 0.7635) were:

$$\alpha_l = \begin{cases} 1, & l = \text{EQUI}, \\ -1, & l = \text{OPPO}, \\ 0.7, & \text{otherwise} \end{cases}$$

Results Following the official evaluation for the SemEval task, we evaluate both alignments and their

[4] http://www.gurobi.com/

Setting	untyped		typed	
	ali	score	ali	score
Baseline	0.8462	0.7610	0.5462	0.5461
Rank 1	0.8194	0.7865	0.7031	**0.6960**
D_A	**0.9257**	0.8377	**0.7350**	0.6776
$D_A + D_S$	0.9235	**0.8591**	0.7281	0.6948

(a) Headlines results

Setting	untyped		typed	
	ali	score	ali	score
Baseline	0.8556	0.7456	0.4799	0.4799
Rank 1	**0.8922**	**0.8408**	0.6867	0.6708
D_A	0.8689	0.7905	0.6933	0.6411
$D_A + D_S$	0.8738	0.8193	**0.7011**	**0.6769**

(b) Imags results

Table 2: F-score for headlines and images datasets. These tables show the result of our systems, baseline and top-ranked systems. D_A is our strong baseline trained on interpretable STS dataset; $D_A + D_S$ is trained on interpretable STS as well as STS dataset. The rank 1 system on headlines is Inspire (Kazmi and Schüller, 2016) and UWB (Konopik et al., 2016) on images. Bold are the best scores.

corresponding similarity scores. The typed alignment evaluation (denoted by **typed ali** in the results table) measures F1 over the alignment edges where the types need to match, but scores are ignored. The typed similarity evaluation (denoted by **typed score**) is the more stringent evaluation that measures F1 of the alignment edge labels, but penalizes them if the similarity scores do not match. The **untyped** versions of alignment and scored alignment evaluations ignore alignment labels. These metrics, based on Melamed (1997), are tailored for the interpretable STS task[5]. We refer the reader to the guidelines of the task for further details. We report both scores in Table 2. We also list the performance of the baseline system (Sultan et al., 2014a) and the top ranked systems from the 2016 shared task for each dataset[6].

By comparing the rows labeled D_A and $D_A + D_S$ in Table 2 (a) and Table 2 (b), we see that in both the headlines and the images datasets, adding sentence level information improves the untyped score, lifting the stricter typed score F1. On the headlines dataset, incorporating sentence-level information degrades both the untyped and typed alignment quality because we cross-validated on the typed score metric.

The typed score metric is the combination of untyped alignment, untyped score and typed alignment. From the row $D_A + D_S$ in Table 2(a), we observe that the typed score F1 is slightly behind that of rank 1 system while all other three metrics are significantly better, indicating that we need to improve our modeling of the intersection of the three aspects. However, this does not apply to images dataset where the improvement on the typed score F1 comes from the typed alignment.

Further, we see that even our base model that only depends on the alignment data offers strong alignment F1 scores. This validates the utility of jointly modeling alignments and chunk similarities. Adding sentence data to this already strong system leads to performance that is comparable to or better than the state-of-the-art systems. Indeed, our final results would have been ranked first on the images task and a close second on the headlines task in the official standings.

The most significant feedback coming from sentence-level information is with respect to the chunk similarity scores. While we observed slight change in the unscored alignment performance, for both the headlines and the images datasets, we saw improvements in both scored precision and recall when sentence level data was used.

6 Analysis and Discussion

In this section, first, we report the results of manual error analysis. Then, we study the ability of our model to handle data from different domains.

6.1 Error Analysis

To perform a manual error analysis, we selected 40 examples from the development set of the headlines section. We classified the errors made by the full model trained on the alignment and sentence datasets. Below, we report the four most significant types of errors:

1. **Contextual implication**: Chunks that are meant to be aligned are not synonyms by them-

[5]In the SemEval 2016 shared task, the typed score is the metric used for system ranking.

[6]http://alt.qcri.org/semeval2016/task2/

selves but are implied by the context. For instance, *Israeli forces* and *security forces* might be equivalent in certain contexts. Out of the 16 instances of EQUI being misclassified as SPE, eight were caused by the features' inability to ascertain contextual implications. This also accounted for four out of the 15 failures to identify alignments.

2. **Semantic phrase understanding**: These are the cases where our lexical resources failed, *e.g.*, *ablaze* and *left burning*. This accounted for ten of the 15 chunk alignment failures and nine of the 21 labeling errors. Among these, some errors (four alignment failures and four labeling errors) were much simpler than others that could be handled with relatively simple features (e.g. *family reunions* $\leftrightarrow$ *family unions*).

3. **Preposition semantics**: The inability to account for preposition semantics accounts for three of the 16 cases where EQUI is mistaken as a SPE. Some examples include *at 91* $\leftrightarrow$ *aged 91* and *catch fire* $\leftrightarrow$ *after fire*.

4. **Underestimated** EQUI **score**: Ten out of 14 cases of score underestimation happened on EQUI label.

Our analysis suggests that we need better contextual features and phrasal features to make further gains in aligning constituents.

6.2 Does the text domain matter?

In all the experiments in Section 5, we used sentence datasets belonging to the same domain as the alignment dataset (either headlines or images). Given that our model can take advantage of two separate datasets, a natural question to ask is how the domain of the sentence dataset influences overall alignment performance. Additionally, we can also ask how well the trained classifiers perform on out-of-domain data. We performed a series of experiments to explore these two questions. Table 3 summarizes the results of these experiments.

The columns labeled Train and Test of the table show the training and test sets used. Each dataset can be either the headlines section (denoted by hdln), or the images section (img) or not used ($\emptyset$). The last two columns report performance on the test set. The rows 1 and 5 in the table correspond to the in-domain settings and match the results of typed alignment and score in Table 2.

Id	Train		Test	Typed F1	
	D_A	D_S		ali	score
1.	hdln	$\emptyset$	hdln	0.7350	0.6776
2.		img		0.6826	0.6347
3.		$\emptyset$	img	0.6547	0.5989
4.		img		0.6161	0.5854
5.	img	$\emptyset$	img	0.6933	0.6411
6.		hdln		0.7033	0.6793
7.		$\emptyset$	hdln	0.6702	0.6274
8.		hdln		0.6672	0.6445

Table 3: F-score for the domain adaptation experiments. This table shows the performance of training on different dataset combinations.

When the headlines data is tested on the images section, we see that there is the usual domain adaptation problem (row 3 vs row 1) and using target images sentence data does not help (row 4 vs row 3). In contrast, even though there is a domain adaptation problem when we compare the rows 5 and 7, we see that once again, using headlines sentence data improves the predicted scores (row 7 vs row 8). This observation can be explained by the fact that the images sentences are relatively simpler and headlines dataset can provide richer features in comparison, thus allowing for stronger feedback from sentences to constituents.

The next question concerns how the domain of the sentence dataset D_S influences alignment and similarity performance. To answer this, we can compare the results in every pair of rows (*i.e.*, 1 vs 2, 3 vs 4, etc.) We see that when the sentence data from the image data is used in conjunction to the headlines chunk data, it invariably makes the classifiers worse. In contrast, the opposite trend is observed when the headlines sentence data augments the images chunk data. This can once again be explained by relatively simpler sentence constructions in the images set, suggesting that we can leverage linguistically complex corpora to improve alignment on simpler ones. Indeed, surprisingly, we obtain marginally better performance on the images set when we use images chunk level data in conjunction

with the headlines sentence data (row 6 vs the row labeled $D_A + D_S$ in the Table 2(b)).

7 Related Work

Aligning words and phrases between pairs of sentences is widely studied in NLP. Machine translation has a rich research history of using alignments (for *e.g.*, (Koehn et al., 2003; Och and Ney, 2003)), going back to the IBM models (Brown et al., 1993). From the learning perspective, the alignments are often treated as latent variables during learning, as in this work where we treated alignments in the sentence level training examples as latent variables. Our work is also conceptually related to (Ganchev et al., 2008), which asked whether improved alignment error implied better translation.

Outside of machine translation, alignments are employed either explicitly or implicitly for recognizing textual entailment (Brockett, 2007; Chang et al., 2010a) and paraphrase recognition (Das and Smith, 2009; Chang et al., 2010a). Additionally, alignments are explored in multiple ways (tokens, phrases, parse trees and dependency graphs) as a foundation for natural logic inference (Chambers et al., 2007; MacCartney and Manning, 2007; MacCartney et al., 2008). Our proposed aligner can be used to aid such applications.

For predicting sentence similarities, in both variants of the task, word or chunk alignments have extensively been used (Sultan et al., 2015; Sultan et al., 2014a; Sultan et al., 2014b; Hänig et al., 2015; Karumuri et al., 2015; Agirre et al., 2015b; Banjade et al., 2015, and others). In contrast to these systems, we proposed a model that is trained jointly to predict alignments, chunk similarities and sentence similarities. To our knowledge, this is the first approach that combines sentence-level similarity data with fine grained alignments to train a chunk aligner.

8 Conclusion

In this paper, we presented the first joint framework for aligning sentence constituents and predicting constituent and sentence similarities. We showed that our predictive model can be trained using both aligned constituent data *and* sentence similarity data. Our jointly trained model achieves state-of-the-art performance on the task of predicting interpretable sentence similarities.

Acknowledgments

The authors wish to thank the anonymous reviewers and the members of the Utah NLP group for their valuable comments and pointers to references.

References

[Agirre et al.2012] Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*.

[Agirre et al.2015a] 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. 2015a. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

[Agirre et al.2015b] Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2015b. UBC: Cubes for English Semantic Textual Similarity and Supervised Approaches for Interpretable STS. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

[Agirre et al.2016a] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016a. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation*.

[Agirre et al.2016b] Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016b. SemEval-2016 Task 2: Interpretable Semantic Textual Similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation*.

[Banjade et al.2015] Rajendra Banjade, Nobal B Niraula, Nabin Maharjan, Vasile Rus, Dan Stefanescu, Mihai Lintean, and Dipesh Gautam. 2015. NeRoSim: A System for Measuring and Interpreting Semantic Textual Similarity. *Proceedings of the 9th International Workshop on Semantic Evaluation*.

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

[Brown et al.1993] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*.

[Chambers et al.2007] Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine De Marneffe, Daniel Ramage, Eric Yeh, and Christopher D Manning. 2007. Learning Alignments and Leveraging Natural Logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics.

[Chang et al.2010a] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010a. Discriminative Learning over Constrained Latent Representations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[Chang et al.2010b] Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser, and Dan Roth. 2010b. Structured Output Learning with Indirect Supervision. In *In Proceedings of the 27th International Conference on Machine Learning*.

[Clarke et al.2012] James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. 2012. An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

[Dagan et al.2013] Ido Dagan, Dan Roth, Mark Sammons, and Fabio M. Zanzotto. 2013. Recognizing Textual Entailment: Models and Applications. *Synthesis Lectures on Human Language Technologies*.

[Das and Smith2009] 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*.

[Dolan et al.2004] Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*.

[Ganchev et al.2008] Kuzman Ganchev, Joao V Graça, and Ben Taskar. 2008. Better Alignments= Better Translations? *Proceedings of ACL-08: HLT*.

[Ganitkevitch et al.2013] 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: Human Language Technologies*.

[Hänig et al.2015] Christian Hänig, Robert Remus, and Xose De La Puente. 2015. ExB Themis: Extensive Feature Extraction from Word Alignments for Semantic Textual Similarity. *Proceedings of the 9th International Workshop on Semantic Evaluation*.

[Karumuri et al.2015] Sakethram Karumuri, Viswanadh Kumar Reddy Vuggumudi, and Sai Charan Raj Chitirala. 2015. UMDuluth-BlueTeam: SVCSTS-A Multilingual and Chunk Level Semantic Similarity System. *Proceedings of the 9th International Workshop on Semantic Evaluation*.

[Kazmi and Schüller2016] Mishal Kazmi and Peter Schüller. 2016. Inspire at SemEval-2016 Task 2: Interpretable Semantic Textual Similarity Alignment based on Answer Set Programming. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, June.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

[Konopik et al.2016] Miloslav Konopik, Ondrej Prazak, David Steinberger, and Tomáš Brychcín. 2016. UWB at SemEval-2016 Task 2: Interpretable Semantic Textual Similarity with Distributional Semantics for Chunks. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, June.

[Leacock and Chodorow1998] Claudia Leacock and Martin Chodorow. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. *WordNet: An Electronic Lexical Database*.

[Lin1998] Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *In Proceedings of the 15th International Conference on Machine Learning*.

[MacCartney and Manning2007] Bill MacCartney and Christopher D Manning. 2007. Natural Logic for Textual Inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

[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 2008 Conference on Empirical Methods in Natural Language Processing*.

[Manning et al.2014] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

[Melamed1997] Dan Melamed. 1997. Manual Annotation of Translational Equivalence: The Blinker Project.

Technical report, Institute for Research in Cognitive Science, Philadelphia.

[Och and Ney2003] Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics, Volume 29, Number 1, March 2003*.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[Resnik1995] Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*.

[Roth and Yih2004] Dan Roth and Wen-Tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*.

[Sultan et al.2014a] 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 of Computational Linguistics*.

[Sultan et al.2014b] 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*.

[Sultan et al.2015] 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*.

[Taskar et al.2004] Ben Taskar, Carlos Guestrin, and Daphne Koller. 2004. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems 16*.

[Tjong Kim Sang and Buchholz2000] Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

[Tsochantaridis et al.2005] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research, Volume 6*.