Neural Network Alignment for Sentential Paraphrases

Jessica Ouyang and **Kathleen McKeown**
Deptartment of Computer Science
Columbia University
New York, NY 10027
{ouyangj, kathy}@cs.columbia.edu

Abstract

We present a monolingual alignment system for long, sentence- or clause-level alignments, and demonstrate that systems designed for word- or short phrase-based alignment are ill-suited for these longer alignments. Our system is capable of aligning semantically similar spans of arbitrary length. We achieve significantly higher recall on aligning phrases of four or more words and outperform state-of-the-art aligners on the long alignments in the MSR RTE corpus.

1 Introduction

Monolingual paraphrase alignment is an active area of research, with applications in many natural language processing tasks, such as text-to-text generation (Barzilay and Elhadad, 2003; Barzilay and McKeown, 2005), natural language inference (MacCartney et al., 2008), and recognizing textual similarity (Sultan et al., 2014b). Madnani and Dorr (2010) identify three levels of paraphrasing. The first is lexical paraphrasing, where individual words are replaced by synonyms or hypernyms. The second, phrasal paraphrasing, involves equivalent idiomatic phrases, such as verb-preposition combinations (eg. "take over" or "assume control of"), or syntactic transformations, such as active versus passive voice.

In this work, we focus on the third: sentential paraphrasing. Sentential paraphrasing can trivially be achieved by performing lexical and phrasal paraphrasing on parts of a sentence, but Madnani and Dorr note that more interesting paraphrases, such as "He needed to make a quick decision in that situation" and "The scenario required him to make a split-second judgment," are challenging.

Past work has focused on lexical and short phrasal alignments, in part because most existing corpora consist of mostly word-level alignments. Yao et al. (2013b) report that 95% of alignments in the MSR RTE (Brockett, 2007) and Edinburgh++ (Cohn et al., 2008) corpora are single-token, lexical paraphrases, and phrases of four or more words are less than 1% of MSR RTE and 3% of Edinburgh++.

In this work, we present a monolingual aligner for long phrasal and sentential paraphrases. Our contributions are as follows:

- Our pointer-network-based system aligns phrases of arbitrary length.
- Our system aligns directly at the phrase level by composing the semantics of the words in each phrase into a single representation of the meaning of the entire phrase.
- We conduct experiments on aligning long paraphrases using the summarization corpus of Ouyang et al. (2017), the first use of this corpus for the alignment task, as well as the MSR RTE corpus (Brockett, 2007).
- We achieve significant increases in recall (over 75 points) while also maintaining a strong lead in F-measure on aligning long paraphrases (involving phrases of four or more words), compared with existing state-of-the-art word- and phrase-based aligners.

2 Related Work

The development of monolingual alignment as an independent natural language processing task began with the release of the Microsoft Research Recognizing Textual Entailment (MSR RTE) corpus (Brockett, 2007), which consists of 1600 sentence pairs, divided evenly into training and testing sets, annotated with alignments. To date, there are only five phrase-based monolingual aligners in existence, not including this work.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4724–4735
Florence, Italy, July 28 - August 2, 2019.

The first aligner developed using the MSR RTE corpus, MANLI (MacCartney et al., 2008), set a precedent for monolingual alignment research: the *possible* alignments in the MSR RTE were not used, following conclusions drawn in machine translation research that training using *possible* alignments does not improve the performance of machine translation systems. As we show in Section 4, this decision, which has been followed by subsequent MSR RTE systems, removed from consideration nearly all of the long alignments (four or more words) in the corpus.

MANLI is a phrase-based system, capable of aligning multiple source tokens to multiple target tokens. However, MacCartney et al. found that constraining it to align only at the word level (ie. setting a maximum phrase length of 1) decreased the system's F-measure by only 0.2%, suggesting that this early work was not yet able to represent the meanings of multi-word phrases as well as it could represent the meanings of single words.

Thadani and McKeown (2011) extended MANLI by introducing syntactic constraints on alignment, improving the system's precision, and used integer linear programming to perform faster, exact decoding, rather than the slower, approximate search used by the original system. Thadani et al. (2012) added dependency arc edits to MANLI's phrase edits, again improving the system's performance. Interestingly, Thadani et al. used both the *sure* and *possible* alignments in the Edinburgh++ corpus (Cohn et al., 2008) and showed that training on both gave better performance than training only on *sure* alignments on this corpus, but no subsequent monolingual alignment systems have taken advantage of *possible* alignments until we do so this work.

The current state-of-the-art phrase-based monolingual alignment system is JacanaAlign-phrase (Yao et al., 2013b), the phrase-based extension of JacanaAlign-token (Yao et al., 2013a). Yao et al. use a semi-Markov CRF to tag each token or sequence of tokens in the source sentence with the indices of aligned target token. To train this system, they synthesized phrasal alignments by merging consecutive lexical alignments among the MSR RTE *sure* alignments; however, even after doing so, they found that long alignments involving phrases of four or more words still made up less than 1% of the corpus. Yao et al. found that the phrase-based JacanaAlign performed slightly worse than the token-based version, likely due to the overwhelming majority of alignments in their test set being at the token level and the token-based annotations in the test set penalizing their phrase-based alignments.

JacanaAlign-phrase is the fastest existing phrase-based aligner (there are only four others: MANLI, its two extensions, and SemAligner, all described in this section), but Yao et al. note that it is roughly 30-60 times slower than JacanaAlign-token. Of particular interest to us is that the decoding time of JacanaAlign-phrase is $\mathcal{O}\left(L_s L_t^2 M N^2\right)$, where L_s and L_t are the maximum allowed phrase lengths, and M and N are the sentence lengths, for the source and target, respectively. The longer the phrases being aligned, the longer Jacana-Align will need to run – we avoid this dependence on phrase length in this work.

Finally SemAligner (Maharjan et al., 2016), like this work, chunks input sentences into phrases before alignment. However, it was designed for and evaluated on the semantic textual similarity task, so its published performance cannot be compared with those of monolingual alignment systems.

3 Models

Our system first chunks the source and target sentences several times, at different levels of granularity, from mostly single words to phrases to whole clauses, then computes a chunk embedding in a distributed semantic space for each chunk (Section 3.1). We call any segmentation of a sentence into chunks a *chunking* of that sentence. We pair each source chunking with each target chunking and use a pointer-network (Vinyals et al., 2015) to perform a preliminary alignment of each source chunk to all target chunks (Section 3.2). Finally, we combine the preliminary alignments from all source/target chunking pairs using a voting system to produce the final alignment from the source sentence to the target sentence (Section 3.3). Implementation details for our model are given in Appendix A in the supplementary material.

3.1 Chunkings and Chunk Embeddings

We chunk the source and target sentences using constituent parsing (Bauer, 2014). We consider all nodes with phrase-level tags (XP) to be *constituents*. Beginning with the leaves, we move up the tree, deleting any node that is wholly contained in a larger constituent but that is neither a con-

I	attended	a wedding	which	offered	no dinner	at	the reception

Figure 1: All potential chunk boundaries.

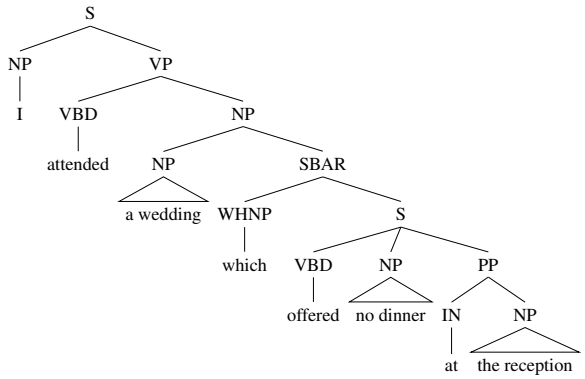

Figure 2: A simplified constituent tree.

stituent itself, nor the sibling of a constituent. Figure 2 shows a simplified constituent tree.

Constituents and their siblings are the smallest possible *chunks* that we consider. In the example constituent tree above, there are eight such small chunks. We can also merge any number of consecutive, small chunks to form a larger chunk: "offered," "no dinner," "at," and "the reception," for instance, can be merged to form "offered no dinner at the reception." In a sentence with i of these smallest chunks, there are $i - 1$ potential chunk boundaries (Figure 1). Since merging two adjacent chunks is equivalent to ignoring the chunk boundary between them, there are 2^{i-1} unique *chunkings* of the sentence. Note that each token in the sentence is contained in only one chunk in each chunking of that sentence.

From the example sentence above, we obtain 128 unique chunkings. The coarsest consists of a single chunk containing the entire sentence, and the most fine-grained has each leaf of the constituent tree as a separate chunk. We do not choose a single chunking to use, but rather represent a sentence by all its possible chunkings. This allows us to align at any level of granularity, from mostly words to full sentences. The multiple chunkings also have the practical benefit of increasing the amount of training data available, with each chunking providing another training instance.

To represent the meaning of a chunk as a whole, we look to recent work in composing word embeddings into phrase- or sentence-level embeddings. Since Mitchell and Lapata (2008), there has been a great deal of interest in learning phrase embeddings (Baroni and Zamparelli, 2010; Zanzotto et al., 2010; Yessenalina and Cardie, 2011; Socher et al., 2012; Grefenstette et al., 2013; Mikolov et al., 2013; Yu and Dredze, 2015). In this work, we generate chunk embeddings using the LSTM language model of Hill et al. (2016)[1]. The model is trained on dictionaries: it takes as input a dictionary definition, in the form of a sequence of word embeddings, and produces as output the embedding of the word to which the definition belongs, thus learning to compose the embeddings of the words into a single embedding representing the entire phrase or sentence. By representing each chunk by a single chunk embedding, we are able to align chunks of arbitrarily large size with only the language model's run time as overhead.

3.2 Preliminary Alignment

For a given source sentence chunking and target sentence chunking, we perform a preliminary alignment using a neural network aligner inspired by the pointer network of Vinyals et al. (2015). Most previous work on neural network alignment used feed-forward, recurrent, or convolutional neural networks to score source-target word pairs and then fed these scores to a traditional alignment model, such as an HMM or a greedy aligner (Yang et al., 2013; Tamura et al., 2014; Legrand et al., 2016), rather than using the neural network itself to predict the alignments. This is due to the difficulty of adapting a neural network to the alignment task directly: two input sequences of unknown and often different lengths, as well as an output set of unknown size.

Our neural network aligner is based on the pointer network and learns a distribution over an output dictionary of variable size. The flexibility of the output size makes the pointer network well-suited to our task of aligning chunkings of variable length. We fix a source chunk from the source chunking under consideration and adapt the pointer network to predict a preliminary alignment over the entire target chunking:

$$a_j^i = v^T \tanh(W_1 e_i + W_2 c_j)$$

where e_i is the embedding for chunk i in the source chunking, c_j is the embedding for candi-

[1]We experimented with averaging word embeddings, but this approach underperformed the language model.

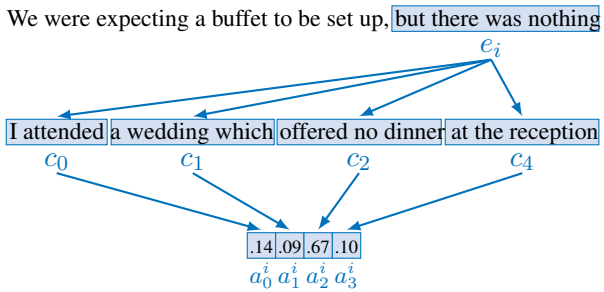

Figure 3: The pointer network performing preliminary alignment a given source chunk and target chunking.

date chunk j in the target chunking, and v, W_1, and W_2 are learned parameters. (For convenience, in subsequent sections we use e_i and c_j to refer to both the chunk embeddings, which are vectors, and to the chunks themselves, which are sequences of tokens.) The chunk embeddings are generated by the LSTM language model described in the previous section, and are fixed at training time. For each source chunk i, the pointer network produces a distribution over all candidate chunks in the target chunking. Figure 3 shows the pointer network aligning a source/target chunking pair.

3.3 Voting and Final Alignment

For a fixed source chunking and a fixed source chunk i, the pointer network produces one preliminary alignment for each unique chunking of the target sentence. We perform this preliminary alignment for all source chunks in all chunkings of the source sentence. By aligning preliminary alignments for all combinations of source and target chunkings, we are able to defer deciding the lengths of the spans we align, instead allowing the voting procedure to discover them.

The final output of our system is aligned token pairs. This is due to our voting procedure, which is described in Figure 4. Because the preliminary alignments are performed on chunkings of different granularities, we must vote at the level of the smallest possible chunks (the leaves in the constituent tree). Since it is not possible for the tokens within one of these smallest possible chunks to receive different amounts of votes (to do so would require the tokens to be in two different chunks in some chunking), and since the standard evaluation for monolingual alignment consists of precision, recall, and F-measure for token pairs – even for phrase-based models – we simply vote on token pairs; each token pair inherits the preliminary alignment value of the source and target chunks

Inputs
- the source sentence W
- the target sentence U
- the set of source sentence chunkings S
- the set of target sentence chunkings T

Initialize
- set score$(w, u) = 0$ for tokens $w \in W$ and $u \in U$

Repeat for $e_i \in s$, for $(s, t) \in S \times T$
- predict preliminary alignment a^i
- add a^i_j to score(w, u) for tokens $w \in e_i$ and $u \in c_j$

Repeat for $w \in W$
- sum-to-one normalize score(w, u) for $u \in U$
- sort pairs (w, u) by score(w, u) in descending order: score$(w, u_1) > \ldots >$ score(w, u_m)
- select max k such that score$(w, u_k) > 1/(k+1)$
- set $A_w = \{(w, u_1), \ldots, (w, u_k)\}$

Return $\bigcup_{w \in W} A_w$

Figure 4: Voting procedure for final output.

containing them. The longer aligned phrases that correspond to these aligned token pairs can be easily constructed: following MacCartney et al. (2008) and Yao et al. (2013b), two tokens are aligned if and only if the phrases containing them are aligned.

Intuitively, only one chunk e_{i_s} in a given source chunking s contains the token w, and only one chunk c_{j_t} in a given target chunking t contains the token u. Here, i_s and j_t indicate the specific source and target chunks that contain the tokens w and u, respectively. The token-level scores are obtained by summing the preliminary alignment values for all source/target chunk pairs where the source chunk contains w and the target chunk contains u:

$$\text{score}(w, u) = \sum_{s \in S} \sum_{t \in T} a^{i_s}_{j_t}$$

where S is the set of all source chunkings of the source sentence, T is the set of all chunkings of the target sentence, and $a^{i_s}_{j_t}$ is the preliminary alignment value described in the previous section.

For a fixed source token w, we normalize its scores to produce a probability distribution over all target tokens. We select the k highest-scoring target tokens such that the score of each token is greater than $1/(k+1)$. If we select four target tokens, for example, each has a score of at least 0.2, and the next-highest-scoring token has a score of less than 0.167. Intuitively, we are looking for a large gap in the target token scores at which to cut off the selected tokens from the unselected tokens; the sum of the scores of all unselected tokens is less than the score of any selected token. We select the largest possible number of tar-

Very rarely do I get a “thanks” or a smile of appreciation. I never get any thanks.	I had a sleep paralysis dream that I was abducted by aliens. I had the alien abduction dream.

Figure 5: Examples of long alignments from Ouyang et al.'s summarization corpus.

Tilda Swinton *has a prominent role as* **the White Witch**.
Tilda Swinton *plays the part of* **the White Witch.**

Figure 6: An MSR RTE pair, slightly edited for length, with sure alignments **bolded** and possible alignments *italicized*.

get tokens for which this requirement holds. This flexible threshold ensures that the selected tokens $u_1, \ldots u_k$ have much larger scores than the unselected tokens $u_{k+1}, \ldots, u_m$ while still allowing any number of tokens to be selected. The selected target tokens are then aligned to the source token w to produce aligned token pairs. The final output of our system is the union of the aligned token pairs for each source token in the source sentence.

4 Data

4.1 The MSR RTE Corpus

The MSR RTE corpus (Brockett, 2007) has been used extensively for training and evaluating alignment systems and consists of mostly word-level alignments. In order to use this corpus to train a phrase-based alignment system, Yao et al. (2013b) created longer alignments by merging consecutive word-level alignments in the MSR RTE training set into larger, phrase-level alignments. They reported that doing so increased the percentage of multi-word alignments from 4% to 21%. However, even after this merging, alignments involving at least one phrase of four words or longer still make up less than 1% of the corpus.

Examining the MSR RTE training set, we find that it does contain some sentence pairs with longer alignments – but these alignments are marked as *possible* (approximate) rather than *sure* (exact). Most aligners designed for this corpus, including MANLI and some of its extensions (MacCartney et al., 2008; Thadani and McKeown, 2011), both word- and phrase-baseed JacanaAlign (Yao et al., 2013a,b), and Sultan et al. (2014a, 2015), are trained and evaluated on the *sure* alignments only[2]. Figure 6 shows a sentence pair containing a *possible* alignment: if only the *sure* alignments are considered, neither of the alignments involves phrases of four or more words, but if *possible* alignments are included, the aligned phrases are much longer.

If we include *possible* alignments, the percentage of alignments in the MSR RTE training set involving phrases of four or more words increases to 27%, and if we restrict ourselves to sentence pairs that contain a *possible* alignment, that percentage increases to 61%. Unfortunately, the MSR RTE training set consists of 800 sentence pairs, a very small amount of data for a neural network, and restricting the sentence pairs to those containing *possible* alignments reduces the amount of data even further. Because of its relatively small size, we do not use the MSR RTE corpus to train our alignment model; however, we evaluate on the subset of 406 sentence pairs in the MSR RTE test set that contain *possible* alignments.

4.2 The Ouyang et al. Corpus

To train our model, we use the narrative summarization corpus of Ouyang et al. (2017), which consists of pairs of abstractive and extractive summaries of online personal narratives. The abstractive summaries in the corpus were written from scratch and aligned back to the original narratives to produce extractive summaries – they are human-written paraphrases. Figure 5 shows two sentence-level alignments from this corpus.

The corpus contains 6173 alignments created by workers on Amazon Mechanical Turk, who were instructed to align “phrases from the [abstractive] summary with phrases from the [narrative] that effectively mean the same things.” The workers were free to align phrases of any length, including the full sentences shown above. Examining these alignments, we find that just over 99% involve phrases of four or more words, and the average length of aligned phrases is 11 for abstractive summary sentences and 25 for extractive summary sentences. This corpus contains a relatively large amount of long alignments, precisely the type of data we need to train our alignment model.

5 Experiments

We report the results of our experiments using the standard alignment evaluation metrics of pre-

[2] Yao (2014) performs experiments using a different definition of “sure” and “possible”: his “sure” alignments are those with perfect agreement among the MSR RTE annotators, and “possible” are those with disagreement.

cision, recall, and F-measure for aligned token pairs, where two tokens are considered aligned if and only the phrases containing them are aligned. As Yao et al. (2013b) argue, evaluating at the token level allows for alignment systems to receive partial credit for phrases that are partially, but not fully, aligned correctly. We do not report the exact match percentage simply because that number was close to zero for all systems we tested – getting an exact match on a long alignment is difficult.

5.1 Baselines

We compare our aligner against three systems: Sultan et al. (2014a), a state-of-the-art word-level aligner; JacanaAlign-phrase (Yao et al., 2013b), a state-of-the-art phrase-based aligner, and SemAligner (Maharjan et al., 2016). As discussed in Section 2, SemAligner has not previously been evaluated as a monolingual alignment system, as it is designed as a textual similarity system, but we include it as a baseline because its approach of aligning chunks is more similar to ours. SemAligner assigns semantic relations to pairs of chunks, so in this evaluation, we treat chunk pairs assigned the *equivalent*, *specification*, and *related* relations as aligned and the *opposite* relation as not aligned. Because the evaluations are on phrase-level alignments, for fairness, we follow Yao et al. in converting word-level alignments into phrase-level ones by merging consecutive single-word alignments into larger phrase alignments.

We also evaluate a greedy baseline on Ouyang et al., which scores each candidate chunk in the target based on the cosine similarity between its phrase embedding and that of the source chunk. We calculate the score using cosine distance as follows: let e and c be the phrase embeddings for the source and candidate chunk, respectively.

$$\text{score} = 1 - \frac{ec}{\|e\|\|c\|} + 0.25m$$

where the constituent mismatch indicator m is a binary indicator that takes the value 0 if the source and candidate chunks are of the same constituent type, and 1 otherwise. This penalty encourages the greedy aligner to align constituents of the same type, but still allows, for example, a verb phrase to be aligned to its nominalized form. The mismatch penalty of 0.25 was tuned on our validation set.

The greedy baseline aligns the source chunk to the target chunk with the lowest score. If there are no target chunks with scores below a gap threshold, the source chunk remains unaligned (we use gap threshold of 0.6, also tuned on our validation set). Following MacCartney et al. (2008), we convert chunk-level alignments to word-level by considering two tokens to be aligned if and only if the chunks containing them are aligned. Finally, we take the union of all token alignments for all chunkings of the source and target sentences.

System	P%	R%	F_1%
Sultan et al.	**76.1**	1.4	2.8
SemAligner	65.7	2.5	5.4
Jacana	59.5	3.9	7.3
greedy	51.4	27.5	35.8
pointer	54.3	**79.5**	**64.5**

Table 1: Performance on Ouyang et al. test set.

5.2 Ouyang et al. Evaluation

Table 1 shows the performance of the pointer-aligner on the Ouyang et al. test set, compared with the three other systems and greedy baseline. Our approach has an order of magnitude improvement in recall and F-measure over existing aligners. The greedy baseline also dramatically improves recall, demonstrating the importance of phrase-level similarity, but is significantly worse than the pointer-aligner that is key to success.

Figure 7 shows alignments from the pointer-aligner and from Jacana, which outperformed Sultan et al. and SemAligner, although it did not outperform the greedy baseline[3]. We see that Jacana produces one longer alignment, shown in green; the pointer-aligner aligns the longest spans, although it seems to have trouble with over-aligning and including some extra words ("which I miraculously" in red) while excluding others that should be aligned ("my boyfriend").

5.3 MSR RTE Evaluation

We evaluate on the MSR RTE corpus, using a majority vote among the three annotators: any alignments that at least two annotators marked as *sure* or *possible* are included. Of the 800 sentence pairs in the MSR RTE test set, only 406 contain *possible* alignments. Because we are interested in evaluating the systems on long alignments, we remove from consideration the 394 sentence pairs that do

[3]The alignments from the other systems are included in Appendix B.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(a) Ouyang et al. gold standard annotation.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(b) Pointer-aligner alignment.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(c) Jacana alignment.

Figure 7: Ouyang et al. alignments. Due to length restrictions, we show only the best-performing baseline, Jacana.

Botswana *is a* **business** *partner of* **De Beers.**
Production at mines operated by Debswana – **Botswana***'s* 50-50 *joint* **venture** *with* **De Beers** – reach 33 million carats**.**

(a) MSR RTE gold standard annotation, with sure alignments in **bold** and possible alignments in *italics*.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(b) Pointer-aligner alignment.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(c) Jacana alignment.

Figure 8: MSR RTE alignments. Due to length restrictions, we show only the best-performing baseline, Jacana.

System	P%	R%	F_1%
Sultan et al.	6.7	3.4	4.4
SemAligner	4.1	6.8	5.1
Jacana	5.2	6.7	5.8
pointer	**23.4**	**47.7**	**31.4**

Table 2: Performance on MSR RTE.

not contain any *possible* alignments. As discussed in Section 4, Yao et al. found that, even after merging consecutive single-word alignments, the *sure* alignments of the MSR RTE consist overwhelmingly of phrases fewer than four words in length. It is not until we add in the *possible* alignments that the percentage of four-word or longer phrases grows to 24% in the MSR RTE test set; when we look only at sentence pairs containing a least one *possible* alignment, the percentage of longer phrases grows to 44%. Thus evaluating only on the 406 sentence pairs that contain at least one *possible* requires systems not only to perform well on longer alignments, but also to avoid sacrificing performance on short alignments.

Figure 8 shows alignments from the pointer-aligner and Jacana on an MSR RTE sentence pair[4]. (Note that the pointer-aligner was trained only on the Ouyang et al. data, and not on any MSR RTE data.) This particular pair was very good for the pointer-aligner because the gold standard alignment is neatly separated out from the rest of the sentence as a parenthetical. Jacana's alignments shown in green and yellow suffer from the same noisy, constituent-breaking boundaries as does the pointer-aligner on sentence pairs less perfectly suited to our approach.

6 Discussion and Limitations

Comparing the gold standard alignments of the MSR RTE corpus with those in Ouyang et al., we see that it is often the case with the Ouyang et al. alignments that one side contains much more information than other. While some MSR RTE alignments have this property (eg. "prominent" in Figure 6), not all do. This is likely a side effect of the Ouyang et al. corpus being intended for summarization – the sentence pairs are composed of an excerpt from a narrative and a human-

[4]The alignments from the other systems are included in Appendix C.

written summary, which by definition compresses the content of the narrative. Further, Ouyang et al.'s alignments were generated by Amazon Mechanical Turk workers, who were instructed to highlight aligned spans. In Figure 7a, we see that the clause "who I had been living with for two years" should probably not be aligned. However, the workers may have found it bothersome to remove the clause (which would require splitting the alignment shown in green into two separate alignments), so the clause remains in Ouyang et al.'s gold standard. Being trained on this data, the pointer-aligner seems to have learned this preference for retaining extra information contained within a larger, more strictly aligned span, such as the word "50-50" in Figure 8b. While it is possible for the pointer-aligner to align a single source phrase to two non-consecutive target phrases, it did not encounter such examples in training and never does so in any of our experiments.

The pointer-aligner has difficulty with clean phrase boundaries, eg. omitting "my boyfriend" but including "which I miraculously" in Figure 7b. Because our system considers the score of a token to be the sum of the scores of the chunks that contain that token, it is possible for words within a constituent to have different scores if there is a potential chunk boundary inside the constituent. In the first sentence of Figure 7, for example, there is a potential chunk boundary between "she repaid me by hooking up with" and "my boyfriend" (because "my boyfriend" is itself a constituent). Thus, there is a chunking where "my boyfriend" is its own, separate chunk, and in the preliminary alignment for that chunking, the pointer-network must have assigned "my boyfriend" a lower score than it did the rest of the chunks. While other, coarser chunkings would have given "my boyfriend" some score, it was apparently not enough to make up the difference, and "my boyfriend" did not accumulate enough score to be included in the final alignment. The exclusion of "my boyfriend" is an error on the part of our system, and it may be worth constraining the system not to break up certain types of constituents, such as prepositional phrases.

We were curious how else chunking might affect our results. Our pointer-aligner aligns chunks rather than individual words, and this may introduce some noise to our alignments. For instance, in the example in Figure 3, the phrase "offered no dinner" is a single chunk. If the gold standard alignment had included only "no dinner" and omitted "offered", the preliminary alignments that used this particular chunking would not be able to match the gold standard alignment because they could not align "no dinner" without also aligning "offered." It is also possible that there are errors in our parses, resulting in chunks that are not syntactic constituents; the Ouyang et al. training data consists of informal texts, which contain misspellings and grammatical mistakes that can cause errors in parsing, and thus in our chunkings.

To determine to what extent this problem might affect our experiments, we provided three human annotators (graduate students in our university's Computer Science Department) with two versions of the Ouyang et al. summary-narrative pairs: one with our phrase chunking boundaries marked, and one without. We asked the annotators to align first the unmarked version, and then the marked version, with the constraint that they should respect the marked boundaries and align either all the words in the chunk, or none of them. Our human annotators achieved substantial agreement ($\kappa = 0.729$).

System	P%	R%	F_1%
Human (free)	73.5	27.1	39.6
Human (chunk)	69.4	30.6	42.5
Human (free, no punct.)	**80.2**	34.5	48.3
Human (chunk, no punct.)	76.3	37.6	50.3
Pointer-Aligner	54.3	**79.5**	**64.5**

Table 3: Comparison of human performance with and without chunk boundaries and sentence-final punctuation.

We evaluated our annotators' performance on the Ouyang et al. test set (Table 3). Being constrained to respect chunk boundaries did lower the humans' precision, but increased their recall and overall performance. Thus, we conclude that incorrect phrase chunk boundaries is not so grave a concern.

We also investigated the humans' relatively low recall, and on inspection found that many of Ouyang et al.'s annotators preferred to align entire clauses or sentences where possible, and tended to be less willing to align fragments of sentences than our three annotators were. Amusingly, Ouyang et al.'s annotators almost always include sentence-final punctuation as part of their alignments, while neither our annotators nor our pointer-aligner do,

and removing such punctuation from consideration results in a substantial improvement to our annotators' performance.

The main limitation of our approach is that it is computationally expensive. We expand each pair of input sentences into multiple chunkings, and the pointer-network runs on each pairing of a source chunk and target chunking. The number of potential chunk boundaries in an input sentence varies roughly with sentence length: if the source sentence has length M, and the target sentence has length N, then there are roughly $M/2$ potential chunk boundaries in the source sentence and $N/2$ in the target. There are then 2^{M-1} unique chunkings of the source sentence and 2^{N-1} of the target. The complexity of our system is thus

$$\mathcal{O}\left((M/2+1)\,2^{M-1}2^{N-1}\right) = \mathcal{O}\left(M2^{M+N-3}\right)$$

Our approach in its current form is not an improvement in complexity over the $\mathcal{O}\left(L_s L_t^2 M N^2\right)$ of Yao et al. (2013b). However, it is important to note that, unlike Yao et al., our system's complexity in no way depends on the lengths of the phrases being aligned, and it can be easily reduced. In the current system, there is a great deal of redundancy among chunkings. Each chunking is identical to one other chunking but for one merge/no merge decision at one potential chunk boundary; thus the preliminary alignments for these chunkings are nearly identical. If instead we fix a constant number of chunkings to align – say the most granular chunking (the leaves of the constituent tree), the second coarsest (the subject and predicate of the sentence), and one more chunking at an intermediate granularity – we sacrifice some flexibility in phrase length but drastically reduce complexity to the much more manageable $\mathcal{O}(M)$.

7 Conclusion

We have presented a pointer-network-based system for aligning longer paraphrases. This pointer-aligner uses an LSTM language model to compose the embeddings of words in a chunk into a chunk embedding and and then aligns these chunks. It is able to align arbitrarily long phrases, automatically discovering the best phrase length, from individual words to full sentences, at which to align a given input sentence pair, and it significantly outperforms existing phrase-based aligners at aligning long phrases with high semantic similarity but low lexical overlap. Our system achieves high recall but suffers from imprecise alignment boundaries. In future work, we intend to refine these alignment boundaries and to optimize the alignment procedure for speed. We hope that this work will raise more interest in developing alignment systems for longer paraphrases.

Acknowledgments

This research is based upon work supported in part by the National Science Foundation under Grant No. IIS-1422863 and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract # FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 Conference on Empirical Mmethods in Natural Language Processing*.

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*.

John Bauer. 2014. Shift-reduce constituency parser. https://nlp.stanford.edu/software/srparser.html. Accessed: 2018-11-30.

Chris Brockett. 2007. Aligning the rte 2006 corpus. Technical report, Microsoft Research.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation*.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the conference on empirical methods in natural language processing*.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*.

Nabin Maharjan, Rajendra Banjade, Nobal B Niraula, and Vasile Rus. 2016. Semaligner: A method and tool for aligning chunks with semantic relation types and semantic similarity scores. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 2008 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jessica Ouyang, Serina Chang, and Kathy McKeown. 2017. Crowd-sourced iterative annotation for narrative summarization corpora. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

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*.

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*.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Feature-rich two-stage logistic regression for monolingual alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Kapil Thadani, Scott Martin, and Michael White. 2012. A joint phrasal and dependency model for paraphrase alignment. In *Proceedings of the 24th International Conference on Computational Linguistics*.

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: Human Language Technologies*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Xuchen Yao. 2014. *Feature-driven Question Answering with Natural Language Alignment*. Ph.D. thesis.

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*.

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 on Natural Language Processing*.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(a) Ouyang et al. gold standard annotation.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(b) Pointer-aligner alignment.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(c) Greedy baseline alignment. The source phrase "hooking up" aligned to both the green and yellow target phrases.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(d) SemAligner alignment.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(e) Jacana alignment.

I saved my friend's life from a heroin overdose, and she repaid me by hooking up with my boyfriend.
I saved my (at the time) best friend from a heroin overdose by sticking suboxone under her tongue, which I miraculously had with me at the time. About a month later her and my boyfriend who I had been living with for two years hooked up.

(f) Sultan et al. alignment.

Figure 9: All Ouyang et al. alignments.

Botswana *is a* **business** *partner of* **De Beers.**
Production at mines operated by Debswana – **Botswana***'s* 50-50 *joint* **venture** *with* **De Beers** – reach 33 million carats.

(a) MSR RTE gold standard annotation, with sure alignments in **bold** and possible alignments in *italics*.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(b) Pointer-aligner alignment.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(c) SemAligner alignment.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(d) Jacana alignment.

Botswana is a business partner of De Beers.
Production at mines operated by Debswana – Botswana's 50-50 joint venture with De Beers – reach 33 million carats.

(e) Sultan et al. alignment.

Figure 10: All MSR RTE alignments.

Appendices

A Implementation Details

Our phrase embedding model is implemented with Lasagne and trained for 25 epochs using the dictionary datasets and hyperparameter settings of Hill et al. Our alignment model (hereafter *pointer-aligner*) is implemented with PyTorch, using the pointer network settings of Vinyals et al. and cosine distance of the predicted alignment a^i from

the gold standard alignment as the loss function. We randomly split Ouyang et al.'s summary pairs into 511 training, 108 validation, and 423 testing pairs, and within each subset further divided each summary pair into sentence pairs. We trained for 16 epochs using early stopping based on validation set performance.

B Full Ouyang et al. Example

The alignments from the pointer-aligner all baseline systems on the example in Figure 7 in the paper are shown on on the next page. While Sultan et al. aligns at the word-level, consecutive alignments that we merged for evaluation are shown in the same color here.

C Full MSR RTE Example

The alignments from the pointer-aligner all three existing alignment systems on the example in Figure 8 in the paper are shown on the next page. While Sultan et al. aligns at the word-level, consecutive alignments that we merged for evaluation are shown in the same color here.