Monolingual Phrase Alignment as Parse Forest Mapping

Sora Kadotani[*]
NTT Human Informatics Laboratories,
NTT Corporation, Japan
sora.kadotani@ntt.com

Yuki Arase
Osaka University, Japan
Artificial Intelligence Research Center,
AIST, Japan
arase@ist.osaka-u.ac.jp

Abstract

We tackle the problem of monolingual phrase alignment conforming to syntactic structures. The existing method formalises the problem as unordered *tree* mapping; hence, the alignment quality is easily affected by syntactic ambiguities. We address this problem by expanding the method to align parse *forests* rather than 1-best trees, where syntactic structures and phrase alignment are simultaneously identified. The proposed method achieves efficient alignment by mapping forests on a packed structure. The experimental results indicated that our method improves the phrase alignment quality of the state-of-the-art method by aligning forests rather than 1-best trees.

1 Introduction

Monolingual phrase alignment, which identifies semantically corresponding phrase pairs in sentences, is a fundamental technique useful for paraphrase recognition (Das and Smith, 2009), textual entailment recognition (MacCartney et al., 2008; Heilman and Smith, 2010), question answering (Wang and Manning, 2010), and interpreting semantic textual similarity (Agirre et al., 2015; Li and Srikumar, 2016). Its ability to declare overlapping information across sentences is also useful for summarisation (Brook Weiss et al., 2021) and for interactive document exploration (Shapira et al., 2017; Hirsch et al., 2021). There are two approaches to phrase alignment: one aligns chunks of arbitrary spans (e.g., n-grams) (Yao et al., 2013; Ouyang and McKeown, 2019; Lan et al., 2021) while the other targets on syntactic phrases (Arase and Tsujii, 2017, 2020). In this study, we take the latter approach to identify phrasal paraphrases conforming to syntactic structures that allow modelling sentences based on syntax (Socher et al., 2013; Tai et al., 2015).

[Person name] made this statement at the Karachi Shipyard when talking with reporters

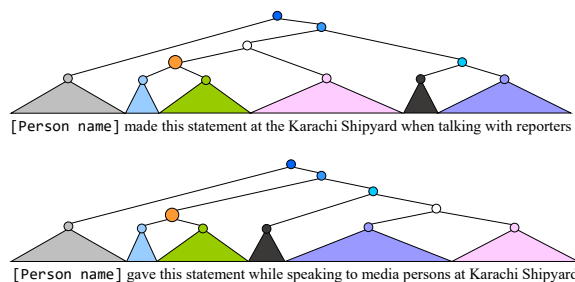[Person name] gave this statement while speaking to media persons at Karachi Shipyard

Figure 1: Phrase alignment example by the proposed method (corresponding nodes are colour-coded). Our method aligns a source (top) and target (bottom) parse forests simultaneously determining their structures.

The current state-of-the-art syntactic phrase alignment (Arase and Tsujii, 2020) has formulated the phrase alignment as the unordered tree mapping problem between trees of source and target sentences.[1] Their method realised an efficient alignment with a solid theoretical background by adopting the constrained tree edit distance algorithm (Zhang, 1996), which aligns syntactic *trees*. However, their experiments were limited to using manually assigned gold syntactic trees, disregarding the effects of syntactic ambiguities that cause parse errors in practical parsers.

We address this problem by expanding the method proposed by Arase and Tsujii (2020) to align parse *forests*. Specifically, our method considers the likelihood of both syntactic structures and phrase alignment, i.e., it simultaneously identifies the syntactic structures of input sentence pairs and phrase alignment within. Figure 1 illustrates phrase alignment by our method, where the trees show syntactic structures of the source (top) and target (bottom). The alignment is colour-coded; the same colour nodes are pairs. For example, the pair of orange nodes represent that verb phrases 'made this statement' and 'gave this statement' are paraphrases. Remarkably, in the source tree, the

---

[*]This work was completed at Osaka University.

[1]We refer to one sentence of a pair as a source and another as a target for the sake of explanation.

449

prepositional phrase 'at the Karachi Shipyard' (the pink node) is correctly attached to the preceding verb phrase (the orange node) because the attachment ambiguity is resolved by referring to the target. In contrast, the 1-best tree failed to derive this structure.

The experimental results on the standard corpus indicated that the proposed method improves the phrase alignment quality of the state-of-the-art aligning 1-best parse trees. We also conducted a manual analysis that revealed attachment errors can be addressed by forest alignment.

2 Preliminary: Tree Alignment

Arase and Tsujii (2020) has formulated phrase alignment as the unordered tree mapping. They adopted the constrained tree edit distance (CTED) algorithm (Zhang, 1996) to identify optimal mappings of phrases in polynomial time. The CTED algorithm is based on dynamic programming; hence their method recursively aligns phrases from leaves to root nodes of source and target syntactic trees.

In the alignment algorithm, alignment of node i and j, denoted as $\langle i, j \rangle$, incurs a cost defined by a function $\gamma(\langle i, j \rangle) \to \mathbb{R}$. In their method, the cost function is the cosine distance between phrase vectors of spans covered by i and j, where the vectors are computed by pooling token representations obtained by a fine-tuned pre-trained language model. A phrase is allowed not to have correspondence, i.e., *null* alignment, which is modelled as alignment to an empty node τ_ϕ. The cost of null alignment is predetermined and given as a hyperparameter λ_ϕ.

We denote T_i as the subtree rooted at node i. If we delete the node i from T_i, there remain a set of subtrees whose root nodes have been the children of i: $\{i_\ell | i_1, \cdots, i_{n_i}\}$, where n_i is the number of the children. When we do not assume the order among these subtrees, they constitute an *unordered* forest, denoted as F_i. The CTED algorithm recursively computes the minimum cost to align subtrees of T_i^s and T_j^t as follows.

$$D(T_i^s, T_j^t) =$$
$$\min \begin{cases} D(\tau_\phi, T_j^t) + \min_{1 \le k \le n_j} \{D(T_i^s, T_{j_k}^t) - D(\tau_\phi, T_{j_k}^t)\}, \\ D(T_i^s, \tau_\phi) + \min_{1 \le \ell \le n_i} \{D(T_{i_\ell}^s, T_j^t) - D(T_{i_\ell}^s, \tau_\phi)\}, \\ D(F_i^s, F_j^t) + \gamma(\langle i, j \rangle). \end{cases} \quad (1)$$

Specifically, Equation (1) computes the minimum cost among the cases regarding the alignment of i and j, i.e., $\langle \tau_\phi, j \rangle$, $\langle i, \tau_\phi \rangle$, and $\langle i, j \rangle$, which correspond to the first, second, and the third expressions,

respectively. Notice that the last case (i.e., $\langle i, j \rangle$) requires the alignment cost of forests under these nodes, i.e., F_i^s and F_j^t. The cost to align F_i^s and F_j^t is computed as follows.

$$D(F_i^s, F_j^t) =$$
$$\min \begin{cases} D(\tau_\phi, F_j^t) + \min_{1 \le k \le n_j} \{D(F_i^s, F_{j_k}^t) - D(\tau_\phi, F_{j_k}^t)\}, \\ D(F_i^s, \tau_\phi) + \min_{1 \le \ell \le n_i} \{D(F_{i_\ell}^s, F_j^t) - D(F_{i_\ell}^s, \tau_\phi)\}, \\ \min_{\mathcal{M}(i,j)} \gamma(\mathcal{M}(i,j)). \end{cases} \quad (2)$$

Here, the first two expressions correspond to null alignment. The third expression identifies the constrained edit distance mapping between F_i^s and F_j^t, denoted as $\mathcal{M}(i, j)$, of the minimum cost. It can be efficiently solved as the minimum cost maximum flow problem on a graph constructed based on these forests. For more details of the alignment algorithm, please refer to (Arase and Tsujii, 2020).

Some phrases may have long-distance correspondences (Heilman and Smith, 2010; Arase and Tsujii, 2017) that cannot be monotonically composed of alignment of descendant nodes, which hence cannot be identified by the CTED algorithm. Arase and Tsujii (2020) align such phrases by heuristic-based post-processing.

3 Proposed Method: Forest Alignment

We expand the alignment method proposed by Arase and Tsujii (2020) to align parse forests instead of trees. The syntactic structures of the input sentence pair are determined simultaneously with phrase alignment. A naive approach to align forests is considering combinations of all candidate trees and then finding the best one. However, this procedure is prohibitively computationally expensive considering the number of valid tree structures. We achieve efficient forest alignment by expanding the CTED algorithm to perform tree mapping on a packed forest structure (Miyao and Tsujii, 2008).

Syntactic Plausibility Studies on parallel parsing (Burkett et al., 2010; Choe and McClosky, 2015) have shown that syntactic ambiguity can be resolved by referring to sentences parsed in parallel with each other. Inspired by these studies, we consider the likelihood of parsing in the alignment cost function. Specifically, Equation (2) is expanded to consider the parsing likelihoods:

$$\hat{D}(F_i^s, F_j^t) = D(F_i^s, F_j^t) - \lambda_s \frac{S(T_i^s) + S(T_j^t)}{2}, \quad (3)$$
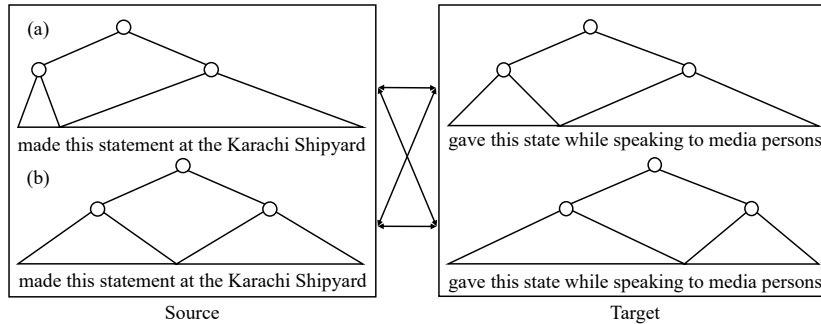
Figure 2: Packed forest structure efficiently stores possible forests under the same nodes.

where $S(\cdot)$ indicates the likelihood of a subtree obtained from a syntactic parser, and λ_s is a hyper-parameter that balances both terms.

Alignment on Packed Forests The packed forest structure corresponds to the packed charts in the CFG parsing and can represent an exponential number of trees with a polynomial number of nodes. Specifically, a packed forests under the node i, $\mathrm{PF}^i = \{F_k^i\}_k$, stores different possible syntactic structures (forests) under i. Figure 2 shows examples. Each box corresponds to a node where different possible structures are stored. In the left box, the source can be composed by combining (a) a verb phrase 'made' and noun phrase 'this statement at the Karachi Shipyard' and (b) a verb phrase 'made this statement' and prepositional phrase 'at the Karachi Shipyard'.

Algorithm 3.1 illustrates our alignment mechanism. It computes the cost to align all combinations of possible structures on the packed forests and memorizes only the one with the minimum cost. That is, only the pair with the minimum cost needs to be considered in the alignment of the upper nodes. In the examples in Figure 2, there are 2 possible structures in the source and target. The proposed method computes the costs of the 2×2 combinations and stores only the minimum cost and the corresponding structures.

4 Experiment

We evaluate the performance of syntactic phrase alignment of the proposed method compared to the previous state-of-the-art.

4.1 Evaluation Corpus

As an evaluation corpus, we used Syntactic Phrase Alignment Dataset for Evaluation (SPADE) (Arase

---

Algorithm 3.1 Packed forest mapping

Input: Packed (unordered) forests under the node i and j: $\mathrm{PF}^{s,i}$ and $\mathrm{PF}^{t,j}$

1: $c_m \leftarrow \infty$ ▷ Minimum cost
2: $\mathrm{MF} \leftarrow \emptyset$ ▷ Pair of forests
3: **for all** $F_k^{s,i} \in \mathrm{PF}^{s,i}$ **do**
4:      **for all** $F_\ell^{t,j} \in \mathrm{PF}^{t,j}$ **do**
5:          $c \leftarrow \hat{D}(F_k^{s,i}, F_\ell^{t,j})$ ▷ Eq. (3)
6:          **if** $c < c_m$ **then**
7:              $c_m \leftarrow c$
8:              $\mathrm{MF} \leftarrow \{F_k^{s,i}, F_\ell^{t,j}\}$
9: Compute $D(T_i^s, T_j^t)$ with c_m, MF ▷ Eq. (1)

---

and Tsujii, 2018).[2] SPADE consists of English paraphrase sentence pairs assigned by their gold constituency trees annotated by linguistic professionals and phrase alignment identified by three native and near-native English speakers. It provides 50 sentence pairs as a development (dev) set and 151 sentence pairs as a test set. While these numbers of sentences may look small, the numbers of phrase pairs are sufficiently large to have statistically meaningful observations, i.e., $8,708$ phrase pairs and $25,709$ phrase pairs in the dev and test sets, respectively. Remind that our method does not require training; only its hyper-parameters should be tuned using the dev set.

4.2 Evaluation Metrics

Metrics for Alignment Quality Alignment recall (**ALIR**), alignment precision (**ALIP**), and alignment F-measure (**ALIF**) are the standard evaluation metrics defined by SPADE. ALIR evaluates how gold-standard alignment can be replicated by automatic alignment, and ALIP measures how automatic alignment overlaps with alignment pairs

---

[2] https://catalog.ldc.upenn.edu/LDC2018T09

identified by at least one annotator.

$$\text{ALIR} = \frac{|\{h|h \in \mathcal{H} \wedge h \in \mathcal{G} \cap \mathcal{G}'\}|}{|\mathcal{G} \cap \mathcal{G}'|}, \qquad (4)$$

$$\text{ALIP} = \frac{|\{h|h \in \mathcal{H} \wedge h \in \mathcal{G} \cup \mathcal{G}'\}|}{|\mathcal{H}|}, \qquad (5)$$

where $\mathcal{H}$ is a set of identified pairs, $\mathcal{G}$ and $\mathcal{G}'$ are those obtained by two respective annotators, and the operator $|\cdot|$ counts the elements in a set. ALIF computes the harmonic mean of ALIR and ALIP. Because SPADE provides alignment pairs by three annotators, there are three combinations for $\mathcal{G}$ and $\mathcal{G}'$. The final ALIR, ALIP, and ALIF values are calculated by taking the averages.

Note that these evaluation metrics count null alignment pairs also; hence, ALIP performs differently from the general precision in that stricter models will have lower ALIP scores. This is because a stricter model aligning only a small number of phrases ($\neq \tau_\phi$) increases the number of null alignment pairs, making $|\mathcal{H}|$ larger.

Metric for Phrase Structure We also evaluated the correctness of phrase structures as the phrase span matching ratio (**PSMR**) against the gold trees. Specifically, PSMR computes the ratio of gold spans that exactly match with the spans in aligned trees. We compute the macro-average of PSMR of all source and target sentences.

4.3 Baseline

We compared our method to the state-of-the-art (Arase and Tsujii, 2020) on the SPADE corpus (denoted as **TreeAligner** hereafter). Their original experiments aligned gold syntactic trees annotated in SPADE. To replicate a realistic scenario where gold syntactic structures are unavailable, we used an off-the-shelf syntactic parser, namely, Enju (Miyao and Tsujii, 2008).[3] We evaluated TreeAligner by inputting the 1-best trees obtained by Enju as the baseline. In contrast, the proposed method (denoted as **ForestAligner** hereafter) takes parse forests in the packed representation obtained by Enju as input.

4.4 Model Settings

For replicating TreeAligner, we used the released codes of the authors.[4] We implemented our ForestAligner based on them using Pytorch[5]. As the

	Structure	ALIR	ALIP	ALIF	PSMR
TreeAligner	Gold tree	88.2	86.6	87.4	100.0
TreeAligner	1-best tree	79.8	76.7	78.2	93.1
ForestAligner	Forest	**81.1**	**79.3**	**80.2**	**93.4**

Table 1: Experimental results on the SPADE test set (%) (the performance of TreeAligner on the gold trees were borrowed from the original paper.)

phrase representation model in both TreeAligner and ForestAligner, we commonly used the bidirectional encoder representations from transformers (BERT) (Devlin et al., 2019) fine-tuned by Arase and Tsujii (2020)[6]. After inputting a sentence pair, a phrase representation was obtained by mean-pooling the token representations consisting of the corresponding phrase.

The hyperparameters were tuned to maximise the evaluation metrics on the SPADE dev set. For TreeAligner, the hyperparameter of the null alignment cost, λ_ϕ, was set to 0.75 to maximise ALIF on the dev set. For ForestAligner, λ_ϕ and λ_s were set to 0.80 and 3.0×10^{11},[7] respectively, to maximise the arithmetic mean of ALIF and PSMR on the dev set.

4.5 Results

Table 1 shows the experimental results. The ALIR, ALIP, and ALIF scores of TreeAligner significantly dropped when aligning 1-best trees compared to the case of aligning gold trees (the first row). Our ForestAligner improved ALIR by 1.3%, ALIP by 2.6%, and ALIF by 2.0% compared to TreeAligner with 1-best trees, which confirms the effectiveness of forest alignment.

For PSMR, ForestAligner moderately improved TreeAligner by 0.3%, which shows the parse errors in 1-best trees can be fixed through forest alignment. To investigate what kind of parse errors were addressed and newly introduced by forest alignment, we randomly sampled 40 sentence pairs where the PSMR score increased (20 sentences) and decreased (20 sentences) compared to TreeAligner. One of the authors observed resultant trees and manually categorised them into error types. Table 2 shows the results, indicating

Error type	Improved	Deteriorated
PP attachment	8	8
NP attachment	5	1
Modifier attachment	1	4
Coordination	2	2
Other	4	5

Table 2: Error analysis of syntactic structures

Human label	$[0, 1)$	$[1, 2)$	$[2, 3)$	$[3, 4)$	$[4, 5]$
Prediction	3.7	3.6	3.6	3.3	4.0

Table 3: Average similarity scores per human labels converted from ForestAligner's alignment costs

ForestAligner tends to fix noun phrase attachment errors while increases modifier attachment errors. The prepositional phrase attachment is a mixture of both improvements and deterioration.

Figure 1 illustrates alignment results by ForestAligner. For the source sentence, the correct structure of composing a phrase 'made this statement at the Karachi Shipyard' with a child verb phrase 'made this statement' (the orange node) and prepositional phrase 'at the Karachi Shipyard' (the pink node) were identified. In contrast, in the 1-best tree, the prepositional phrase was wrongly attached to a noun phrase of 'this statement' to compose a phrase 'this statement at the Karachi Shipyard', which prevented alignment of 'made this statement' and 'gave this statement' (the orange node pair).

5 Discussion: Alignment of Less-Similar Sentences

As discussed in Section 1, phrase alignment is coveted by various applications like paraphrase and textual entailment recognition and question answering. Such applications are different from SPADE, i.e., alignment of paraphrases, in that they require alignment of *less-similar* sentences, too. It is not a trivial difference as it sounds.

As a preliminary experiment, we aligned the test set[8] of the semantic textual similarity (STS) benchmark (Cer et al., 2017) and converted alignment costs into similarity scores. Specifically, we normalised the root-level alignment costs by sentence lengths[9] and scaled them to be compatible with the STS labels, i.e., from 0 (dissimilar) to 5 (equivalent). As a result, Pearson's correlation coefficient of the predicted scores and human labels was limited to 0.51, which is comparable to estimating sentence-level similarity using static word embeddings.[10]

Table 3 shows the average similarity scores per human label. While ForestAligner outputs a noticeably high score on the most similar sentence pairs, other scores are almost uniform on less similar sentences. We conjecture that one of the factors causing this phenomenon is the lack of exposure to less-similar examples during development. The same can happen on existing phrase alignment methods trained on annotated corpora consisting of paraphrasal or highly similar sentence pairs (Thadani et al., 2012; Lan et al., 2021). The distributions of alignment pairs are largely different in semantically similar and less-similar sentences, where alignment is dense in the former but sparse in the latter. Hence, alignment methods trained only on similar sentences may tend to align phrases that should be unaligned.

While there are only a few corpora annotating alignment on less-similar sentences (Ernst et al., 2021), this direction is worth exploring to apply alignment techniques in practical applications. In future work, we will create corpora of this kind and explore robust phrase alignment on both similar and less-similar sentences.

Acknowledgements

---

[8]We excluded 7 sentence pairs that Enju failed to output.
[9]Alignment costs obviously depend on sentence lengths.
[10]https://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 252–263.

Yuki Arase and Junichi Tsujii. 2017. Monolingual phrase alignment on parse forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–11.

Yuki Arase and Junichi Tsujii. 2018. SPADE: Evaluation dataset for monolingual phrase alignment. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Yuki Arase and Jun'ichi Tsujii. 2020. Compositional phrase alignment and beyond. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1611–1623.

Daniela Brook Weiss, Paul Roit, Ayal Klein, Ori Ernst, and Ido Dagan. 2021. QA-align: Representing cross-text content overlap by aligning question-answer propositions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9879–9894.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 127–135.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 1–14.

Do Kook Choe and David McClosky. 2015. Parsing paraphrases with joint inference. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1223–1233.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 468–476.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Ori Ernst, Ori Shapira, Ramakanth Pasunuru, Michael Lepioshkin, Jacob Goldberger, Mohit Bansal, and Ido Dagan. 2021. Summary-source proposition-level alignment: Task, datasets and supervised baseline. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 310–322.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1011–1019.

Eran Hirsch, Alon Eirew, Ori Shapira, Avi Caciularu, Arie Cattan, Ori Ernst, Ramakanth Pasunuru, Hadar Ronen, Mohit Bansal, and Ido Dagan. 2021. iFacetSum: Coreference-based interactive faceted summarization for multi-document exploration. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 283–297.

Wuwei Lan, Chao Jiang, and Wei Xu. 2021. Neural semi-Markov CRF for monolingual word alignment. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 6815–6828.

Tao Li and Vivek Srikumar. 2016. Exploiting sentence similarities for better alignments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2193–2203.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 802–811.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics (CL)*, 34(1):35–80.

Jessica Ouyang and Kathy McKeown. 2019. Neural network alignment for sentential paraphrases. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4724–4735.

Ori Shapira, Hadar Ronen, Meni Adler, Yael Amsterdamer, Judit Bar-Ilan, and Ido Dagan. 2017. Interactive abstractive summarization for event news tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 109–114.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1556–1566.

Kapil Thadani, Scott Martin, and Michael White. 2012. A joint phrasal and dependency model for paraphrase alignment. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1229–1238.

Mengqiu Wang and Christopher Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1164–1172.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Semi-Markov phrase-based monolingual alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 590–600.

Kaizhong Zhang. 1996. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(3):205–222.