Semi-Markov Phrase-based Monolingual Alignment

Xuchen Yao and Benjamin Van Durme
Johns Hopkins University
Baltimore, MD, USA

Chris Callison-Burch∗
University of Pennsylvania
Philadelphia, PA, USA

Peter Clark
Allen Institute for Artificial Intelligence
Seattle, WA, USA

Abstract
We introduce a novel discriminative model for phrase-based monolingual alignment using a semi-Markov CRF. Our model achieves state-of-the-art alignment accuracy on two phrase-based alignment datasets (RTE and paraphrase), while doing significantly better than other strong baselines in both non-identical alignment and phrase-only alignment. Additional experiments highlight the potential benefit of our alignment model to RTE, paraphrase identification and question answering, where even a naive application of our model’s alignment score approaches the state of the art.

1 Introduction
Various NLP tasks can be treated as an alignment problem: machine translation (aligning words in one language with words in another language), question answering (aligning question words with the answer phrase), textual entailment recognition (aligning premise with hypothesis), paraphrase detection (aligning semantically equivalent words), etc. Even though most of these tasks involve only a single language, alignment research has primarily focused on the bilingual setting (i.e., machine translation) rather than monolingual. Moreover, most work has considered token-based approaches over phrase-based.1 Here we seek to address this imbalance by proposing better phrase-based models for monolingual word alignment.

Most token-based alignment models can extrinsically handle phrase-based alignment to some extent. For instance, in the case of NYC aligning to New York City, the single source word NYC may align three times separately to the target words: NYC→New, NYC→York, NYC→City. Or in the case of identical alignment, New York City aligning to New York City is simply New→New, York→York, City→City. However, it is not as clear how to token-align New York (as a city) with New York City. The problem is more prominent when aligning phrasal paraphrases or multiword expressions, such as pass away and kick the bucket. This suggests an intrinsically phrase-based alignment model.

The token aligner jacana-align (Yao et al., 2013a) has achieved state-of-the-art result on the task of monolingual alignment, based on previous work of Blunsom and Cohn (2006). It employs a Conditional Random Field (Lafferty et al., 2001) to align tokens from the source sentence to tokens in the target sentence, by treating source tokens as “observation” and target tokens as “hidden states”. However, it is not designed to handle phrase-based alignment, largely due to the Markov nature of the underlying model: a state can only span one token each time, making it unable to align multiple consecutive tokens (i.e. a phrase). We extend this model by introducing semi-Markov states for phrase-based alignment: a state can instead span multiple consecutive time steps, thus aligning phrases on the source side. Also, we merge phrases on the target side to phrasal states, allowing the model to align phrases on the target side as well. We evaluate the resulting semi-Markov

1Performed while faculty at Johns Hopkins University.

1In this paper we use the term token-based alignment for one-to-one alignment and phrase-based for non one-to-one alignment, and word alignment in general for both.
CRF model on the task of phrase-based alignment, and then show a basic application in the NLP tasks of recognizing textual entailment, paraphrase identification, and question answering sentence ranking. The final phrase-based aligner is open-source.\footnote{http://code.google.com/p/jacana/}

## 2 Related Work

Most work in monolingual alignment employs dependency tree/graph matching algorithms, including tree edit distance (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Heilman and Smith, 2010; Yao et al., 2013b), Particle Swarm Optimization (Mehdad, 2009), linear regression/classification models (Chambers et al., 2007; Wang and Manning, 2010), and min-cut (Roth and Frank, 2012). These works inherently only support token-based alignment, with phrase-like alignment achieved by first merging tokens to phrases as a preprocessing step.

The MANLI aligner (MacCartney et al., 2008) and its derivations (Thadani and McKeown, 2011; Thadani et al., 2012) are the first known phrase-based aligners specifically designed for aligning English sentence pairs. It applies discriminative perceptron learning with various features and handles phrase-based alignment of arbitrary phrase lengths. MANLI suffers from slow decoding time due to its large search space. This was optimized by Thadani and McKeown (2011) through Integer Linear Programming (ILP), where benefiting from modern ILP solvers they showed an order-of-magnitude speedup in decoding. Also, various syntactic constraints can be easily added, significantly improving exact alignment match rate for whole sentence pairs. Besides the common application of textual entailment and question answering, monolingual alignment has also been applied in the field of text generation (Barzilay and Lee, 2003; Pang et al., 2003).

Word alignment has been more explored in machine translation. The IBM models (Brown et al., 1993) allow many-to-one alignment and are essentially asymmetric. Phrase-based MT historically relied on heuristics (Koehn, 2010) to merge two sets of word alignment in opposite directions to yield phrasal alignment. Later, researchers explored non-heuristic phrase-based methods. Among them, Marcu and Wong (2002) described a joint probability model that generates both the source and target sentences simultaneously. All possible pairs of phrases in both sentences are enumerated and then pruned with statistical evidence. Deng and Byrne (2008) explored token-to-phrase alignment based on HMM models (Vogel et al., 1996) by explicitly modeling the token-to-phrase probability and phrase lengths. However, the token-to-phrase alignment is only in one direction: each target state still only spans one source word, and thus alignment on the source side is limited to tokens. Andrés-Ferrer and Juan (2009) extended the HMM-based method to Hidden Semi-Markov Models (HSMM) (Ostdorf et al., 1996), allowing phrasal alignments on the source side. Finally, Bansal et al. (2011) unified the HSMM models with the alignment by agreement framework (Liang et al., 2006), achieving phrasal alignment that agreed in both directions.

Despite successful usage of generative semi-Markov models in bilingual alignment, this has not been followed with models in discriminative monolingual alignment. Essentially monolingual alignment would benefit more from discriminative models with various feature extractions (just like those defined in MANLI) than generative models without any predefined feature (just like how they were used in bilingual alignment). To combine the strengths of both semi-Markov models and discriminative training, we propose to use the semi-Markov Conditional Random Field (Sarawagi and Cohen, 2004), which was first used in information extraction to tag continuous segments of input sequences and outperformed conventional CRFs in the task of named entity recognition. We describe this model in the following section.

## 3 The Alignment Model

Our objective is to define a model that supports phrase-based alignment of arbitrary phrase length. In this section we first describe a regular CRF model that supports one-to-one token-based alignment (Blunsom and Cohn, 2006; Yao et al., 2013a), then extend it to phrase-based alignment with the semi-Markov model.
3.1 Token-based Model

Given a source sentence $s$ of length $M$, and a target sentence $t$ of length $N$, the alignment from $s$ to $t$ is a sequence of target word indices $a$, where $a_i \in [1, M] \in [0, N]$. We specify that when $a_i = 0$, source word $s_i$ is aligned to a NULL state, i.e., deleted. This models a many-to-one alignment from source to target: multiple source words can be aligned to the same target word, but not vice versa. One-to-many alignment can be obtained by running the aligner in the other direction. The probability of alignment sequence $a$ conditioned on both $s$ and $t$ is then:

$$p(a \mid s, t) = \frac{\exp\left(\sum_{i,k} \lambda_{i,k}(a_{i-1}, a_i, s, t)\right)}{Z(s, t)}$$

This assumes a first-order Conditional Random Field (Lafferty et al., 2001). Since the word alignment task is evaluated over $F_1$, instead of directly optimizing it, we choose a much easier objective (Gimpel and Smith, 2010) and add a cost function to the normalizing function $Z(s, t)$ in the denominator:

$$Z(s, t) = \sum_{a} \exp\left(\sum_{i,k} \lambda_{i,k}(\hat{a}_{i-1}, \hat{a}_i, s, t) + \text{cost}(a_y, \hat{a})\right)$$

where $a_y$ is the true alignments. $\text{cost}(a_y, \hat{a})$ can be viewed as special “features” that encourage decoding to be consistent with true labels. It is only computed during training in the denominator because in the numerator $\text{cost}(a_y, a_y) = 0$. Hamming cost is used in practice without learning the weights (i.e., uniform weights). The more inconsistency there is between $a_y$ and $\hat{a}$, the more penalized is the decoding sequence $\hat{a}$ through the cost function.

3.2 Phrase-based Model

The token-based model supports $1 : 1$ alignment. We first extend it in the direction of $l_s : 1$, where a target state spans $l_s$ words on the source side ($l_s$ source words align to 1 target word). Then we extend it in the direction of $1 : l_t$, where $l_t$ is the target phrase length a source word aligns to ($1$ source word aligns to $l_t$ target words). The final combined model supports $l_s : l_t$ alignment. Throughout this section we use Figure 1 as an illustrative example, which shows phrasal alignment between the source sentence: (Shops are closed up for now until March) and the target sentence: (Shops are temporarily closed down).

$1 : 1$ alignment is a special case of $l_s : 1$ alignment where the target side state spans $l_s = 1$ source word, i.e., at each time step $i$, the source side word

![Figure 1: A semi-Markov phrase-based model example and the desired Viterbi decoding path. Shaded horizontal circles represent the source sentence (Shops are closed up for now until March) and hollow vertical circles represent the hidden states with state IDs for the target sentence (Shops are temporarily closed down). State 0, a NULL state, is designated for deletion. One state (e.g. state 3 and 15) can span multiple consecutive source words (a semi-Markov property) for aligning phrases on the source side. States with an ID larger than the target sentence length indicate “phrasal states” (states 6-15 in this example), where consecutive target tokens are merged for aligning phrases on the target side. Combining the semi-Markov property and phrasal states yields for instance, a $2 \times 2$ alignment between closed up in the source and closed down in the target.](image)
s_i aligns to one state a_i and the next aligned state a_{i+1} only depends on the current state a_i. This is the Markovian property of the CRF. When l_s > 1, during the time frame [i, i + l_s), all source words \([a_i, a_{i+l_s})\) share the same state a_i. Or in other words, the state a_i “spans” the following l_s time steps. The Markovian property still holds “outside” the time frame l_s, i.e., a_i+l_s only depends on a_i, the previous state l_s time steps ago. But “within” the time frame l_s, the Markovian property does not hold any more: \([a_i, ..., a_{i+l_s-1})\) are essentially the same state a_i. This is the semi-Markov property. States can be distinguished by this property into two types: semi-Markovian states and Markovian states.

We have generalized the regular CRF to a semi-Markov CRF. Now we define it by generalizing the feature function:

\[
p(a \mid s, t) = \frac{\exp(\sum_{i,k,l_s} \lambda_k f_k(a_{i-l_s}, a_i, s, t))}{Z(s, t)}
\]

At time i, the k-th feature function \(f_k\) mainly extracts features from the pair of source words \((s_{i-l_s}, ..., s_{i})\) and target word \(t_{a_i}\) (still with a special case that \(a_i = 0\) marks for deletion). Inference is still Viterbi-like: except for the fact during maximization, the Viterbi algorithm not only checks the previous one time step, but all \(l_s\) time steps. Suppose the allowed maximal source phrase length is \(L_s\), define \(V_i(a \mid s, t)\) as the highest score along the decoding path until time i ending with state a:

\[
V_i(a \mid s, t) = \max_{a_1, a_2, ..., a_{i-1}} p(a_1, a_2, ..., a_i = a \mid s, t)
\]

then the recursive maximization is:

\[
V_i(a \mid s, t) = \max_{a'} \max_{l_s = 1 ... L_s} [V_{i-l_s}(a' \mid s, t) + \Psi_i(a', a, l_s, s, t)]
\]

with factor:

\[
\Psi_i(a', a, l_s, s, t) = \sum_k \lambda_k f_k(a'_{i-l_s}, a_i, s, t)
\]

and the best alignment a can be obtained by backtracking the last state \(a_M\) from \(V_M(a_M \mid s, t)\).

Training a semi-Markov CRF is very similar to the inference, except for replacing maximization with summation. The forward-backward algorithm should also be used to dynamically compute the normalization function \(Z(s, t)\). Compared to regular CRFs, a semi-Markov CRF has a decoding time complexity of \(O(L_s M N^2)\), a constant factor \(L_s\) (usually 3 or 4) slower.

To extend from 1 : 1 alignment to 1 : \(l_t\) alignment with one source word aligning to \(l_t\) target words, we simply explode the state space by \(L_t\) times with \(L_t\) the maximal allowed target phrase length. Thus the states can be represented as an \(N \times L_t\) matrix. The state at \((j, l_t)\) represents the target phrase \([t_{j_2}, ..., t_{j_l})\). In this paper we distinguish states by three types: NULL state \((j = 0, l_t = 0)\), token state \((l_t = 1)\) and phrasal state \((l_t > 1)\).

To efficiently store and compute these states, we linearize the two dimensional matrix with a linear function mapping uniquely between the state id and the target phrase offset/span. Suppose the target phrase \(t_j\) of length \(l_{t_j}\) \([1, L_t]\) holds a position \(p_{t_j} \in [1, N]\), and the source word \(s_i\) is aligned to this state \((p_{t_j}, l_{t_j})\), a tuple for (position, span). Then state id \(a_{s_i}\) is computed as:

\[
a_{s_i}(p_{t_j}, l_{t_j}) = \begin{cases} 
  p_{t_j} & l_{t_j} = 1 \\
  N + (p_{t_j} - 1) \times L_t + l_{t_j} & 1 < l_{t_j} \leq L_t 
\end{cases}
\]

Assume in Figure 1, \(L_t = 2\), then the state id for the phrasal state \((5, 2)\) closed down with \(p_{t_j} = 5\) for the position of word \(\text{down}\) and \(l_{t_j} = 2\) for the span of 2 words (looking “backward” from the word \(\text{down}\)) is: \(5 + (5 - 1) \times 2 + 2 = 15\).

Similarly, given a state id \(a_{s_i}\), the original target phrase position and length can be recovered through integer division and modulation. Thus during decoding, if one output state is 15, we would know that it uniquely comes from the phrasal state \((5,2)\), representing the target phrase \(\text{closed down}\).

This two dimensional definition of state space expands the number of states from \(1 + N\) to \(1 + L_t N\). Thus the decoding complexity becomes \(O(M(L_t N)^2) = O(L_t^2 M N^2)\) with a usual value of 3 or 4 for \(L_t\).

Now we have defined separately the \(l_s : 1\) model and the \(1 : l_t\) model. We can simply merge them to
have an $l_s : l_t$ alignment model. The semi-Markov property makes it possible for any target states to align phrases on the source side, while the two dimensional state mapping makes it possible for any source words to align phrases on the target side. For instance, in Figure 1, the phrasal state $a_{15}$ represents the two-word phrase closed down on the target side, while still spanning for two words on the source side, allowing a $2 \times 2$ alignment. State $a_{15}$ is phrasal, and at source word position 3 and 4 (spanning closed up) it is semi-Markovian. The final decoding complexity is $O(L_s L_t^2 MN^2)$, a factor of $30 \sim 60$ times slower than the token-based model (with a typical value of 3 or 4 for $L_s$ and $L_t$).

In the following we describe features.

### 3.3 Feature Design

We reused features in the original token-based model based on string similarity, POS tags, position, WordNet, distortion and context. Then we used an additional chunker to mark phrase boundaries only for feature extraction:

** Chunking Features ** are binary indicators of whether the phrase types of two phrases match. Also, we added indicators for mappings between source phrase types and target phrase types, such as “vp2np”, meaning that a verb phrase in the source is mapped to a noun phrase in the target.

Moreover, we introduced the following lexical features:

**PPDB Features (Ganitkevitch et al., 2013) ** include various similarity scores derived from a paraphrase database with 73 million phrasal and 8 million lexical paraphrases. Various paraphrase conditional probability was employed. For instance, for the ADJP/VP phrase pair capable of and able to, there are the following minus-log probabilities:

$$p(lhs|e1) = 0.1, p(lhs|e2) = 0.3, p(e1|lhs) = 5.0$$

$$p(e1|e2) = 1.3, p(e2|lhs) = 6.7, p(e2|e1) = 2.8$$

$$p(e1|e2, lhs) = 0.6, p(e2|e1, lhs) = 2.3$$

where $e1/e2$ are the phrase pair, and $lhs$ is the left hand side syntactic non-terminal symbol. We did not use the syntactic part (e.g., the NP of NNS $\leftrightarrow$ the NNS of NP) of PPDB as we did not make the assumption that the input sentence pairs were well-formed (and newswire-like) English, or even of a language with a parser available. Also, for phrasal alignments, we ruled out those paraphrases spanning multiple syntactic structures, or of different syntactic structures (indicated as [X] in PPDB), for instance, and crazy $\leftrightarrow$ mad.

**Semantic Relatedness Feature ** is a single scaled number in $[0, 1]$ from the best performing system (Han et al., 2013) of the *Sem 2013 Semantic Textual Similarity (STS) task. We included this feature mainly to deal with cases where “related” words cannot be well measured by either paraphrases or distributional similarities. For instance, in one alignment dataset annotators aligned married with wife. Adding a few other words as comparison, the Han et al. (2013) system gives the following similarity scores:

- married/wife: 0.85
- married/husband: 0.84
- married/child: 0.10
- married/stone: 0.01

**Name Phylogeny Feature (Andrews et al., 2012) ** is a similarity feature with a string transducer to model how one name evolves to another. Examples below show how similar is the name Bill associated with other names in log probability:

- Bill/Bill: -0.8
- Bill/Billy: -5.2
- Bill/William: -13.6
- Bill/Mary: -18.6

Finally, one decision we made during feature design was not to use any parsing-based features, with a permissive assumption that the input might not be well-formed English, or even not complete sentences (such as fragmented snippets from web search). The “deepest” linguistic processing stays at the level of tagging and chunking, making the model more easily extendable to other languages.

### 3.4 Feature Value

In this phrase-based model, the width of a state span over the source words depends on the competition between features fired on the phrases as a whole vs. the consecutive but individual tokens. We found it critical to assign feature values “fairly” among tokens and phrases to make sure that semi-Markov states and phrasal states fire up often enough for phrasal alignments.
Table 1: Statistics of the two manually aligned corpora, divided into training and test in sentence pairs. The length column shows average lengths of source and target sentences in a pair. %align. is the percentage of aligned tokens.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>train</th>
<th>test</th>
<th>length</th>
<th>%align.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR06</td>
<td>800</td>
<td>800</td>
<td>29/11</td>
<td>36%</td>
</tr>
<tr>
<td>Edinburgh++</td>
<td>715</td>
<td>305</td>
<td>22/22</td>
<td>78%</td>
</tr>
</tbody>
</table>

Table 2: Percentage of various alignment sizes (undirectional, e.g., 1x2 and 2x1 are merged) after synthesizing phrasal alignment from token alignment in the training portion of two corpora.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>1x1</th>
<th>1x2</th>
<th>1x3</th>
<th>2x2</th>
<th>2x3</th>
<th>3x3</th>
<th>more</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR06</td>
<td>89.2</td>
<td>1.9</td>
<td>0.3</td>
<td>5.7</td>
<td>0.0</td>
<td>1.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Edinburgh++</td>
<td>81.9</td>
<td>3.5</td>
<td>0.8</td>
<td>8.3</td>
<td>0.4</td>
<td>3.0</td>
<td>2.1</td>
</tr>
</tbody>
</table>

To illustrate this in a simplified way, take closed up↔closed down in Figure 1, and assume the only feature is the normalized number of matching tokens in the pair. Then this feature firing on the following pairs would have values (the normalization factor is the maximal phrase length):

- closed↔closed: 1.0
- closed up↔closed: 0.5
- closed up↔up: 0.5
- closed up↔closed down: 0.5
- ...↔...

The desired alignment closed up↔closed down would not have survived the state competition due to its weak feature value. In this case the model would simply prefer a token alignment closed↔closed and up↔... (probably NULL).

Thus we upweighted feature values by the maximum source or target phrase length to encourage phrasal alignments, in this case closed up ↔ closed down:1.0. Then this alignment would have a better chance to be picked out with additional features, such as with the PPDB and Semantic Relatedness Features, which are also upweighted by maximum phrase lengths.

4 Experiment

4.1 Data Preparation

There are two annotated datasets for training and testing. **MSR06** (Brockett, 2007) has annotated alignments on the 2006 PASCAL RTE2 development and test corpora, with 1600 pairs in total. **Edinburgh++** (Thadani et al., 2012) is a revised version of the Edinburgh paraphrase corpus (Cohn et al., 2008) with sentences from the following resources: 1. the Multiple-Translation Chinese corpus; 2. Jules Verne’s novel Twenty Thousand Leagues Under the Sea. 3. the Microsoft Research paraphrase corpus (Dolan et al., 2004). The corpus is more balanced and symmetric: the source and target sentences are both 22 words long on average. Table 1 shows some statistics.

Semantically equivalent words and phrases in the premise and hypothesis sentences are aligned in a manner analogous to alignments in statistical machine translation. This dataset is asymmetric: on average the premises contain 29 words and the hypotheses 11 words. **Edinburgh++** is a revised version of the Edinburgh paraphrase corpus (Cohn et al., 2008) with sentences from the following resources: 1. the Multiple-Translation Chinese corpus; 2. Jules Verne’s novel Twenty Thousand Leagues Under the Sea. 3. the Microsoft Research paraphrase corpus (Dolan et al., 2004). The corpus is more balanced and symmetric: the source and target sentences are both 22 words long on average. Table 1 shows some statistics.

Both corpora contain mostly token-based alignment. For MSR06, MacCartney et al. (2008) showed that setting the allowable phrase size to be greater than one only increased $F_1$ by 0.2%. For Edinburgh++, the annotation guideline explicitly instructs to “prefer smaller alignments whenever possible”. Statistics shows that single token alignment counts 96% and 95% of total alignments in these two corpora separately. With such a heavy imbalance towards only token-based alignment, a phrase-based aligner would learn feature weights that award token alignments more than phrasal alignments.

Thus we synthesized phrasal alignments from continuous monotonic token alignments in these two corpora. We first ran the OpenNLP chunker through the corpora. Then for each phrase pair, if each token in the source phrase is aligned to a token in the target phrase in a monotonic way, and vice versa, we

---

3http://www.cs.biu.ac.il/~nlp/files/RTE_2006_Aligned.zip
4http://www.ling.ohio-state.edu/~scott/edinburgh-plusplus
5http://staffwww.dcs.shef.ac.uk/people/T.Cohn/paraphrase_guidelines.pdf
merge these alignments to form one single phrasal alignment.\(^6\) Table 2 lists the percentage of various alignment sizes after the merge. Two observations can be made: first, the portion of phrasal alignments increases to \(10\% \sim 20\%\) after merging; second, allowing a maximal phrase length of 3 covers \(98\% \sim 99\%\) of total alignments, thus a phrase length larger than 3 would be a bad trade-off for coverage vs speed.

4.2 Baselines and Evaluation Metrics
MacCartney et al. (2008) and Yao et al. (2013a) showed that the traditional MT bilingual aligner GIZA++ (Och and Ney, 2003) presented weak results on the task of monolingual alignment. Thus we instead used four other strong baselines:

**Meteor** (Denkowski and Lavie, 2011): a system for evaluating machine translation by aligning MT output with reference sentences. It is designed for the task of monolingual alignment and supports phrasal alignment. We used version 1.4 and default weights to optimize by maximum accuracy.

**MANLI-constraint** (Thadani and McKeown, 2011): a re-implemented MANLI system with ILP-powered decoding for speed and hard syntactic constraints to boost exact match rate, with reported numbers on MSR06.

**MANLI-joint** (Thadani et al., 2012): an improved version of MANLI-constraint that not only models phrasal alignments, but also alignments between dependency arcs, with reported numbers on the original Edinburgh paraphrase corpus.

**jacana-token** (Yao et al., 2013a): a token-based aligner with state-of-the-art performance on MSR06.

Note that the jacana-token aligner is open-source, so we were able to re-train it with exactly the same feature set used by our phrase-based model. This allows a fair comparison of model performance (token-based vs. phrase-based). The MANLI* systems are not available, thus we only report their numbers from published papers.

The standard evaluation metrics for alignments are precision (P), recall (R), \(F_1\), and exact matching rate (E) based on either tokens (two tokens are considered aligned iff they are aligned) or phrases (two tokens are considered aligned iff they are contained within phrases that are aligned). Following Thadani et al. (2012), we only report the results based on token alignments (which allows a partial credit if their containing phrases are not aligned), even for the phrase-based alignment task. The reasoning is that if a phrase-based aligner is already doing better than a token aligner in terms of token alignment scores, then the difference in terms of phrase alignment scores will be even larger. Thus showing the superiority of token alignment scores is sufficient.

4.3 Implementation and Training
The elements in the phrase-based model: dynamic state indices, semi-Markov and phrasal states, are not typically found in standard CRF implementations. Thus we implemented the phrase-based model in the Scala programming language, which is fully interoperable with Java, using one semi-Markov CRF package\(^7\) as a reference. We used the L2 regularizer and LBFGS for optimization. OpenNLP\(^8\) provided the POS tagger and chunker and JWNL\(^9\) interfaced with WordNet (Fellbaum, 1998).

4.4 Results
Table 3 gives scores (in bigger fonts) of different aligners on MSR06 and Edinburgh++ and their corresponding phrasal versions. Overall, the token-based aligner did the best on the original corpora, in which single token alignment counts more than \(95\%\) of total alignment. The phrase-based aligner did slightly worse. We think the main reason was that it output more phrasal alignment, which in turn harms scores in token-based evaluation (for instance, if the gold alignment is New York, then the phrasal alignment of New York would only have half the precision because it inherently also aligns New in the source with York in the target.). Further investigation showed that on the Edinburgh++ corpus, over-generated phrase-based alignment, when evaluated under just token alignment, contributed hurting about \(1.1\%\) of overall \(F_1\),

\(^{6}\text{a few examples: two Atlanta-based companies↔two Atlanta companies, the UK↔the UK, the 17-year-old↔the teenager, was held↔was held.}\)

\(^{7}\text{http://crf.sf.net}\)

\(^{8}\text{http://opennlp.apache.org/}\)

\(^{9}\text{http://jwordnet.sf.net/}\)
a gap that would make the phrase aligner (85.9%) outperform the token aligner (86.4%).

On the phrasal alignment corpora (represented by MSR06p and EDB++p in Table 3), the phrase-based aligner did significantly better. Note that the overall $F_1$ and exact match rate are still much lower than those scores obtained from the original corpora, suggesting that the phrasal corpora present a much harder task. Furthermore, as a more “fair” comparison between the two aligners, we synthesized phrasal alignments from the output of the token-based aligner, just as how the phrase-based corpora were prepared, then evaluated its performance again. Still, on the EDB++p corpus, the token aligner was about 1.6% (current difference is 69.1% vs. 72.8%) worse than the phrase-based aligner.

Also, we want to emphasize that since the token-based aligner and the phrase-based aligner shared exactly the same features and lexical resources, the performance boost of the phrase-based aligner on the phrasal corpora results from a better model design: it is the semi-Markov property and phrasal states making the phrase-based aligner better.

To further investigate the performance of aligners with respect to different types of alignment, we divided the scores into those for identical alignments (such as New↔New) and non-identical alignments (such as wife↔spouse), indicated by the subscripts $i$ and $n$ in Table 3. In terms of identical alignment, most aligners were able to score more than 90%, but for non-identical alignment there was noticeable decrease. Still, on the phrasal alignment corpora, the phrase-based model has a much larger recall score for non-identical alignment than others.

We also divided scores with respect to token-only alignment and phrase-only alignment. Due to space limit, we only show results on synthesized Edinburgh++, in Table 4. Meteor and the token aligner inherently have either very limited or no support for phrasal alignment, thus they had very low scores on phrase-only alignment. We then ran the aligners in two directions and merged the results with the “union” MT heuristic to get better phrase support. But that still did not bring $F_{1p}$’s up to over 5%.

The phrase-based aligner baseline Meteor did worse than our aligners. We think there are two reasons: First, Meteor was not trained on these corpora. Second, Meteor only does strict word, stem, synonym and paraphrase matching but does not use any string similarity measures: this can be supported by the large difference between, for instance, $F_{1i}$ and $F_{1n}$. In general Meteor did well on identical alignment, but not so well on non-identical alignment.

### Table 3: Results on original (mostly token) and phrasal (P) alignment corpora, where (%x%) indicates how much alignment is identical alignment, such as New↔New. E% stands for exact (perfect) match rate. Subscript $i$ stands for corresponding scores for “identical” alignment and $n$ for “non-identical”. *: scores of MANLI-jnt were for the original Edinburgh corpus instead of Edinburgh++ (with hand corrections) so it is not a direct comparison.

<table>
<thead>
<tr>
<th>System</th>
<th>P%</th>
<th>R%</th>
<th>F1%</th>
<th>E%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_{1i}$</td>
<td>$R_{1i}$</td>
<td>$F_{1i}$</td>
<td></td>
</tr>
<tr>
<td>MSR06 (78.6%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meteor</td>
<td>82.5</td>
<td>81.2</td>
<td>81.9</td>
<td>15.0</td>
</tr>
<tr>
<td>MANLI-cons.</td>
<td>89.5</td>
<td>86.2</td>
<td>87.8</td>
<td>33.0</td>
</tr>
<tr>
<td></td>
<td>93.6</td>
<td>83.5</td>
<td>88.3</td>
<td>32.1</td>
</tr>
<tr>
<td></td>
<td>92.1</td>
<td>82.8</td>
<td>86.8</td>
<td>29.1</td>
</tr>
<tr>
<td>MSR06p (69.0%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meteor</td>
<td>82.5</td>
<td>68.3</td>
<td>74.7</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>89.9/39.9</td>
<td>97.3/24.6</td>
<td>93.5/30.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96.6/77.7</td>
<td>96.9/35.6</td>
<td>96.8/48.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95.7/65.0</td>
<td>95.9/34.7</td>
<td>95.8/45.2</td>
<td></td>
</tr>
<tr>
<td>EDB++p (51.7%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meteor</td>
<td>88.3</td>
<td>80.5</td>
<td>84.2</td>
<td>12.7</td>
</tr>
<tr>
<td>MANLI-jnt*</td>
<td>76.6</td>
<td>83.8</td>
<td>79.2</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td>91.3</td>
<td>82.0</td>
<td>86.4</td>
<td>15.0</td>
</tr>
<tr>
<td></td>
<td>90.4</td>
<td>81.9</td>
<td>85.9</td>
<td>13.7</td>
</tr>
<tr>
<td>EDB++++p (51.7%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meteor</td>
<td>88.4</td>
<td>60.6</td>
<td>71.9</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>94.0/61.9</td>
<td>97.8/24.1</td>
<td>95.9/34.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96.4/63.9</td>
<td>97.4/36.4</td>
<td>96.9/46.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>96.0/57.4</td>
<td>97.8/38.3</td>
<td>96.9/46.0</td>
<td></td>
</tr>
</tbody>
</table>

5 Applications

Natural language alignment can be applied to various NLP tasks. While how to most effectively apply
Table 4: Same results on the phrasal Edinburgh++ corpus but with scores divided by token-only alignment (subscript t) and phrase-only alignment (subscript p).

<table>
<thead>
<tr>
<th>System</th>
<th>P%</th>
<th>R%</th>
<th>F1%</th>
<th>E%</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDB++P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>token</td>
<td>59.4/21.4</td>
<td>85.5/0.9</td>
<td>70.1/1.7</td>
<td>2.3</td>
</tr>
<tr>
<td>phrase</td>
<td>82.3</td>
<td>70.4/44.2</td>
<td>72.8</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Table 5: Results (Accuracy, Precision, Recall, Mean Average Precision, Mean Reciprocal Rank) on the tasks of RTE, PP and QA.

<table>
<thead>
<tr>
<th>System</th>
<th>MAP</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cui et al. (2005)</td>
<td>0.4271</td>
<td>0.5259</td>
</tr>
<tr>
<td>Wang et al. (2007)</td>
<td>0.6029</td>
<td>0.6852</td>
</tr>
<tr>
<td>Heilman and Smith (2010)</td>
<td>0.6091</td>
<td>0.6917</td>
</tr>
<tr>
<td>Yao et al. (2013b)</td>
<td>0.6307</td>
<td>0.7477</td>
</tr>
<tr>
<td>the token aligner</td>
<td>0.5982</td>
<td>0.6582</td>
</tr>
<tr>
<td>our phrasal aligner</td>
<td>0.6165</td>
<td>0.7333</td>
</tr>
</tbody>
</table>

6 Conclusion

We have introduced a phrase-to-phrase alignment model based on semi-Markov Conditional Random Fields. The combination of semi-Markov states and phrasal states makes phrasal alignment on both the source and target sides possible. The final phrase-
based aligner performed the best on two phrasal alignment corpora and showed its potential usage in three NLP tasks. Future work includes aligning discontinuous (gappy) phrases and integrating alignment more closely in NLP applications.

Acknowledgement

We thank Vulcan Inc. for funding this work. We also thank Jason Smith, Travis Wolfe, Frank Ferraro and the three anonymous reviewers for their comments and suggestion.

References


