A Recipe for Creating Multimodal Aligned Datasets for Sequential Tasks

Angela S. Lin ♣* **Sudha Rao**♢ **Asli Celikyilmaz**♢ **Elnaz Nouri**♢
Chris Brockett♢ **Debadeepta Dey**♢ **Bill Dolan**♢
♣Salesforce Research, Palo Alto, CA, USA
♢Microsoft Research, Redmond, WA, USA
angela.lin@salesforce.com {sudhra, aslicel, elnouri}@microsoft.com
{chrisbkt, dedey, billdol}@microsoft.com

Abstract

Many high-level procedural tasks can be decomposed into sequences of instructions that vary in their order and choice of tools. In the cooking domain, the web offers many partially-overlapping text and video recipes (i.e. procedures) that describe how to make the same dish (i.e. high-level task). Aligning instructions for the same dish across different sources can yield descriptive visual explanations that are far richer semantically than conventional textual instructions, providing commonsense insight into how real-world procedures are structured. Learning to align these different instruction sets is challenging because: a) different recipes vary in their order of instructions and use of ingredients; and b) video instructions can be noisy and tend to contain far more information than text instructions. To address these challenges, we first use an unsupervised alignment algorithm that learns pairwise alignments between instructions of different recipes for the same dish. We then use a graph algorithm to derive a joint alignment between multiple text and multiple video recipes for the same dish. We release the MICROSOFT RESEARCH MULTIMODAL ALIGNED RECIPE CORPUS[1] containing ∼150K pairwise alignments between recipes across 4,262 dishes with rich commonsense information.

1 Introduction

Although machine learning has seen tremendous recent success in challenging game environments such as Go (Schrittwieser et al., 2019), DOTA (OpenAI, 2019), and StarCraft (DeepMind, 2019), we have not seen similar progress toward algorithms that might one day help humans perform everyday tasks like assembling furniture, applying makeup, repairing an electrical problem, or cooking a particular dish. In part this is because the relevant large-scale multimodal (language, video, audio) datasets are difficult to acquire, even with extensive crowdsourcing (Salvador et al., 2017; Sanabria et al., 2018). Unimodal data, though, is abundant on the web (e.g. instructional videos or textual instructions of tasks). Using language as the link between these modalities, we present an approach for learning large-scale alignment between multimodal procedural data. We hope our work, and the resulting released dataset, will help spur research on real-world procedural tasks.

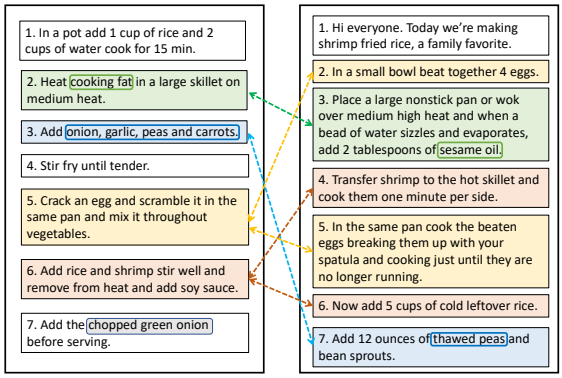

Figure 1: Text recipe (left) and transcript of video recipe (right) for ***shrimp fried rice***. Aligned instructions are highlighted in the same color. Ingredients that can be substituted are encircled in the same color.

Recipes in the cooking domain provide procedural instruction sets that are captured – in large volume – both in video and text-only forms. Instruction sets in these two modalities overlap sufficiently to allow for an alignment that reveals interestingly different information in the linguistic and visual realms. In Figure 1, for instance, the text recipe (left) and the transcribed video recipe (right) for *shrimp fried rice* vary in word usage, order of instructions and use of ingredients. Know-

*Work done when the author was an intern at Microsoft.
[1] https://github.com/microsoft/multimodal-aligned-recipe-corpus

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4871–4884
July 5 - 10, 2020.

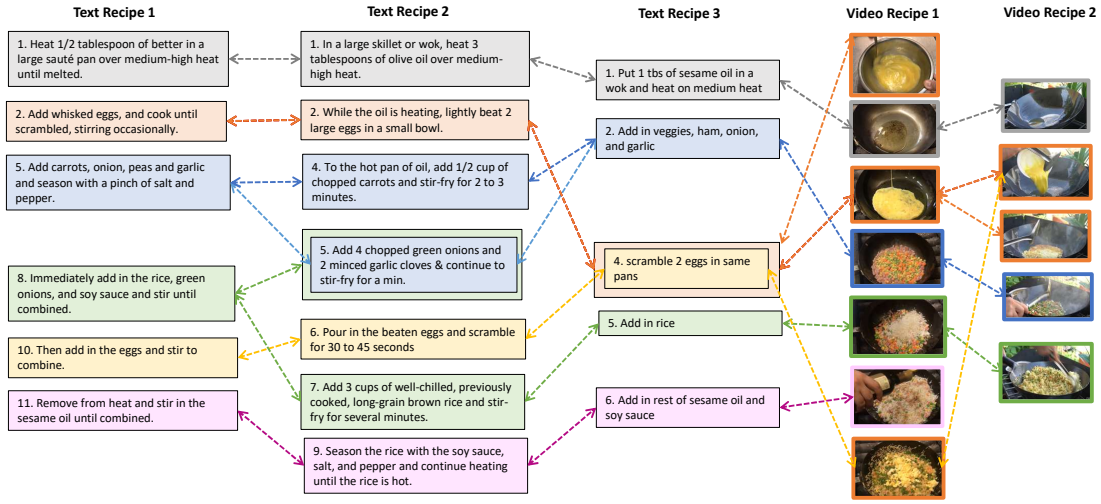

Figure 2: Dish level alignment between three text recipes and two video recipes for ***fried rice***. Same colored text boxes (in text recipes) and image borders (in video recipes) indicate instructions that are aligned to each other.

ing that the highlighted instructions correspond to the same step is useful in understanding potential ingredient substitutions, how the same step can be linguistically described and physically realized in different ways, and how instruction order can be varied without affecting the outcome.

Motivated by this idea that aligned procedural data can be a powerful source of practical commonsense knowledge, we describe our approach for constructing the MICROSOFT RESEARCH MULTIMODAL ALIGNED RECIPE CORPUS. We first extract a large number of text and video recipes from the web. Our goal is to find joint alignments between multiple text recipes and multiple video recipes for the same dish (see Figure 2). The task is challenging, as different recipes vary in their order of instructions and use of ingredients. Moreover, video instructions can be noisy, and text and video instructions include different levels of specificity in their descriptions. Most previous alignment approaches (Munteanu and Marcu, 2005) deal with pairwise alignments. Since our goal is to align multiple instruction sets, we introduce a novel two-stage unsupervised algorithm. In the first stage, we learn pairwise alignments between two text recipes, two video recipes, and between a text and a video recipe using an unsupervised alignment algorithm (§3.1). In the second stage, we use the pairwise alignments between all recipes within a dish to construct a graph for each dish and find a maximum spanning tree of this graph to derive joint alignments across multiple recipes (§3.2).

We train our unsupervised algorithm on 4,262 dishes consisting of multiple text and video recipes per dish. We release the resulting pairwise and joint alignments between multiple recipes within a dish for all 4,262 dishes, along with commonsense information such as textual and visual paraphrases, and single-step to multi-step breakdown (§5).

We evaluate our pairwise alignment algorithm on two datasets: 1,625 text-video recipe pairs across 90 dishes from the YouCook2 dataset (Zhou et al., 2018a), and a small set of 200 human-aligned text-text recipe pairs across 5 dishes from Common Crawl. We compare our algorithm to several textual similarity baselines and perform ablations over our trained model (§4). Finally, we discuss how this data release will help with research at the intersection of language, vision, and robotics (§6).

2 Recipe Data Collection

We describe our approach for collecting large-scale text and video recipes; and constructing recipe pairs for training our unsupervised alignment algorithm.

2.1 Common Crawl Text Recipes

We extract text recipes from Common Crawl,[2] one of the largest web sources of text. We heuristically filter the extracted recipes[3] to obtain a total of 48,852 recipes across 4,262 dishes. The number

[2]https://commoncrawl.org/
[3]Details in supplementary.

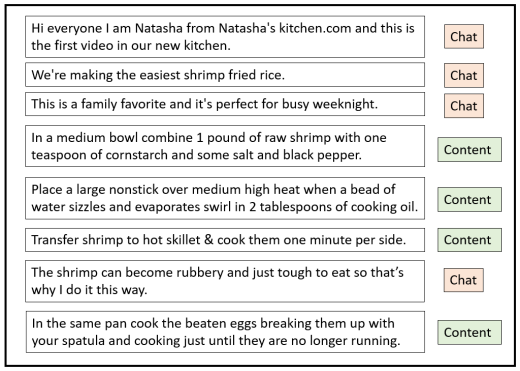

Figure 3: An example transcript of a video recipe with sentences marked as "chat" (non-instructional) or "content" (instructional).

of recipes per dish ranges from 3 to 100 (with an average of 6.54 and standard deviation of 7.22). The average recipe length is 8 instructions.

2.2 YouTube Video Recipes

For each dish in the text recipes, we use the dish name with 'recipe' appended, e.g. 'chocolate chip cookie recipe', as a query on YouTube and extract the top N videos where N is proportional to the number of text recipes for that dish[4] to obtain a total of 77,550 video recipes. We transcribe these videos using the Microsoft Speech-to-Text Cognitive service.[5]

Video recipes, unlike text recipes, contain non-instructional ("chat") information. For instance, the presenter may give an introduction either of themselves or of the dish at the beginning of the video before diving into the steps of the recipe. Figure 3 contains an example transcript with "chat" and "content" information marked. We hypothesize that it is useful to remove such chat information from the transcripts before aligning them to text recipes. We build a supervised chat/content classifier using the YouCook2 dataset (Zhou et al., 2018a), an existing instructional cooking video dataset where parts of video that correspond to instructions are annotated by humans. We assume that these parts correspond to content whereas the rest of the video corresponds to chat.[6] We preprocess the transcriptions of all 77,550 videos using this chat/content classifier[7] to remove all sentences classified as chat.

[4]Details in supplementary.

[5]https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

[6]Details in supplementary.

[7]Classifier achieves 85% F1-score on a held out test set.

	Train	Val	Test
No. of dishes	4,065	94	103
Text-Text Pairs	46,054	5,822	11,652
Text-Video Pairs	56,291	3,800	5,341
Video-Video Pairs	19,200	274	514

Table 1: Statistics of our recipe pairs data (2.3)

2.3 Recipe Pairs for Training

Given N text recipes and M video recipes for a dish, we pair each text recipe with every other text recipe to get $O(N^2)$ text-text recipe pairs. Similarly, we pair each text recipe with every video recipe to get $O(N * M)$ text-video recipe pairs, and pair each video recipe with every other video recipe to get $O(M^2)$ video recipe pairs. On closer inspection, we find that some of these pairs describe recipes that are very different from one other, making a reasonable alignment almost impossible. For example, one black bean soup recipe might require the use of a slow cooker, while another describes using a stove. We therefore prune these recipe pairs based on the match of ingredients and length[8] to finally yield a set of 63,528 text-text recipe pairs, 65,432 text-video recipe pairs and 19,988 video-video recipe pairs. We split this into training, validation and test split at the dish level. Table 1 shows the number of dishes and pairs in each split.

3 Recipe Alignment Algorithm

We first describe our unsupervised pairwise alignment model trained to learn alignments between text-text, text-video, and video-video recipes pairs. We then describe our graph algorithm, which derives joint alignments between multiple text and video recipes given the pairwise alignments.

3.1 Pairwise Alignments between Recipes

Our alignment algorithm is based on prior work (Naim et al., 2014) that learns to align a sequence of natural language instructions to segments of video recording of the same wet lab protocol. They first identify the nouns in the text sentences and the blobs (i.e. objects) in video segments. Given the blobs from M video segments $F = [\mathbf{f}^{(1)}, ..., \mathbf{f}^{(M)}]$ and the nouns from N sentences $E = [\mathbf{e}^{(1)}, ..., \mathbf{e}^{(N)}]$, the task is to learn alignments between video segments and text sentences. They propose a hierarchical generative model which first uses a Hidden Markov Model

[8]Details in supplementary.

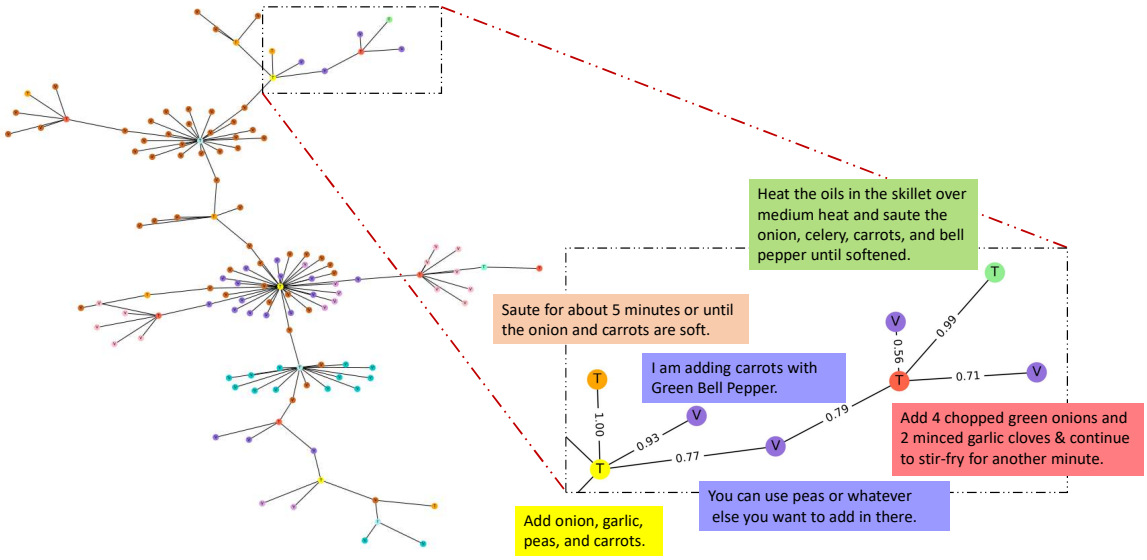

Figure 4: A maximum span tree for ***fried rice*** dish with text instructions and transcript segments as nodes, alignments as edges, and alignment probabilities as edge weights. Nodes representing text instructions are labeled "T". Nodes representing transcript segments are labeled "V". Each color indicates a different recipe. The bounding box shows a magnified section of the tree with edge weights and the instruction/transcript associated with each node.

(HMM) (Rabiner, 1989; Vogel et al., 1996) to generate each video segment $f^{(m)}$ from one of the text sentences $e^{(n)}$. They then use IBM1 model (Brown et al., 1993) emission probabilities to generate the blobs $\{f_1^{(m)}, ..., f_J^{(m)}\}$ in $f^{(m)}$ from the nouns $\{e_1^{(n)}, ..., e_I^{(n)}\}$ in $e^{(n)}$ as follows:

$$P(\mathbf{f}^{(m)}|\mathbf{e}^{(n)}) = \frac{\epsilon}{(I)^J} \prod_{j=1}^{J} \sum_{i=1}^{J} p(f_j^{(m)}|e_i^{(n)}) \quad (1)$$

The hidden state in the HMM model corresponds to the alignment between video segment and text sentence, and the state transition probabilities correspond to the jump between adjacent alignments. For computational tractability, a video segment can be aligned to only one sentence (multiple sentences can align to the same video segment)

We use this algorithm to learn pairwise alignments between text-text, text-video and video-video recipes. Given two recipes (*source* and *target*) of the same dish, we define our alignment task as mapping each text instruction (or video transcript sentence) in the *source* recipe to one or more text instructions (or video transcript sentences) in the *target* recipe.

We make two modifications to the alignment algorithm described above: First, our recipe pairs, unlike the wet lab protocol data, does not follow the same temporal sequence. The alignment algorithm must thus learn to jump within a longer range. We set the window of jump probabilities at $[-2, 2]$.[9] Second, we use transcriptions to learn alignments rather than the objects detected in videos. We hypothesize that the richness of language used in instructional videos may facilitate better alignment with transcripts (as others have observed (Malmaud et al., 2015; Sener et al., 2015)). We use all words (except stop words) in video transcript sentences and all words in text instructions while learning the IBM1 word level probabilities. An instruction in one recipe can be aligned to multiple instructions in the other recipe.

3.2 Joint Alignment among Multiple Recipes

We use the pairwise alignments to derive a joint alignment at the dish level between multiple text and video recipes. For each dish, we construct a graph where each node represents an instruction from a text recipe or a transcript sentence from a video recipe. We use the pairwise alignments to draw edges between nodes, with alignment probabilities as the edge weights. We include only those edges that have alignment probability greater than 0.5. The pairwise alignments are directed since they go from the source recipe to the target recipe.

[9] We find that increasing the window beyond 5 decreases performance.

We first convert the directed graph into an undirected graph by averaging the edge weights between two nodes and converting directed edges into undirected edges. Note that the resultant graph can have multiple connected components as some recipe pairs may not have any instructions aligned with probability greater than the threshold of 0.5

Our goal is to find a set of jointly-alignable instructions across different recipes. We therefore convert the graph (with cycles) into a forest by running the maximum spanning tree algorithm on the graph. Figure 4 shows an example tree derived for one of the dishes. A path in this tree, that has at most one node from each recipe, constitutes a set of jointly-alignable instructions. For example, in the magnified section of the tree in Figure 4, all unique colored nodes in the path from the yellow node to the green node constitute a set of jointly-alignable instructions.

4 Experimental Results

We describe how we evaluate our pairwise alignment algorithm (from §3.1). We answer the following research questions using our experimentation:

1. How does our alignment model perform when evaluated on human-aligned recipe pairs?
2. Does our unsupervised alignment model outperform simpler non-learning baselines?
3. How does performance differ when we use only nouns or nouns and verbs instead of all words to learn alignments?

4.1 Human Aligned Evaluation Set

We evaluate our pairwise alignment algorithm on the following two human annotated datasets:

YouCook2 text-video recipe pairs The YouCook2 dataset (Zhou et al., 2018a) consists of 1,625 cooking videos paired with human-written descriptions for each video segment. These span 90 different dishes. We transcribe all videos using the Microsoft Speech-to-Text Cognitive service[10] and separate it into sentences using a sentence tokenizer. Given a sequence of human-written descriptions and a sequence of transcript sentences, the alignment task is to align each transcript sentence to one of the human-written descriptions. We train our pairwise alignment model on the train split of our text-video recipe pairs (from §2.3) and evaluate on the YouCook2 dataset. An important difference between the text-video pairs in YouCook2 and in our data is that in YouCook2, the text instructions and the video segments are temporally aligned since the text instructions were specifically written for the videos. In our data, however, the text and the video recipes can differ in order.

CommonCrawl text-text recipe pairs We randomly choose 200 text-text recipes pairs (spanning 5 dishes) from the test split of our data (§2.3) and collect alignment annotations for them using six human experts. We show annotators a numbered list of the instructions for the *target* recipe (along with its title and ingredients). We display instructions for the *source* recipe with input boxes besides them and ask annotators to write in the number(s) (i.e labels) of one or more *target* instruction(s) with which it most closely aligns. Each recipe pair is annotated by three annotators. For 65% of the instructions, two or more annotators agree on a label. For only 42% of the instructions do all three annotators agree, suggesting that the difficulty level of this annotation task is high. We train our pairwise alignment model on the train split of our text-text recipe pairs (§2.3) and evaluate on the 200 human-aligned pairs.

4.2 Baselines

Baselines described below align each instruction [11] in the *source* recipe to one or more instructions in the *target* recipe.

Random We align each instruction in the *source* recipe to a random instruction in the *target* recipe.

Uniform alignment Given N instructions in the *target* recipe, we divide the instructions in the *source* recipe into N equal chunks and align each instruction in the i^{th} chunk of the *source* recipe to the i^{th} instruction in the *target* recipe. For instance, given a *source* recipe $[S1, S2, S3, S4]$ and a *target* recipe $[T1, T2]$, uniform alignment would align $S1$ and $S2$ to $T1$ and $S3$ and $S4$ to $T2$. More generally, we align the i^{th} instruction in the *source* recipe to the $[(\frac{N}{M}i)^{th} - (\frac{N}{M}(i+1))^{th})$ instruction in the *target* recipe.

BM25 retrieval We use BM25 (Robertson et al., 2009) as our information retrieval baseline. Given

[10] https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

[11] We use the term "instruction" to mean both text instruction and transcript sentence.

Methods	Precision	Recall	F1
Random	18.53	14.47	14.49
Uniform alignment	63.44	50.81	53.10
BM25 retrieval	48.86	39.85	38.91
Textual Similarity			
Exact word match	46.75	40.70	40.06
TF-IDF	46.82	39.23	38.55
GloVe	46.13	38.74	37.14
BERT	48.83	41.48	40.89
RoBERTa	50.21	42.43	42.28
HMM+IBM1			
Nouns	78.63	63.83	65.29
Nouns+Verbs	80.56	67.90	69.00
All words	**81.39**	**69.27**	**70.30**

Table 2: Results for text-video recipe alignments on YouCook2 dataset.

Methods	Precision	Recall	F1
Random	14.26	14.00	12.69
Uniform alignment	41.38	31.85	33.22
BM25 retrieval	50.06	**55.27**	49.30
Textual Similarity			
Exact word match	53.90	48.39	46.98
TF-IDF	52.78	46.82	45.12
GloVe	56.04	51.89	50.30
BERT	50.72	55.07	49.10
RoBERTa	52.49	**55.86**	50.44
HMM+IBM1			
Nouns	62.11	48.99	50.73
Nouns+Verbs	64.72	50.76	52.97
All words	**66.21**	52.42	**54.55**

Table 3: Results for text-text recipe alignment on Common Crawl dataset.

a source and a target recipe pair, we construct a corpus using all instructions in the target recipe. We then use each source instruction as a query to retrieve the top most instruction from the target instruction corpus and align the source instruction to the retrieved target instruction.

Textual similarity Given a *source* recipe instruction and a *target* recipe instruction, we define a measure of textual similarity between the two instructions using the following five methods. For each *source* instruction, we compute its similarity score with every *target* instruction and align it to the *target* instruction with the highest score.

a. Exact word match: Given two instructions, we define exact word match as the ratio of the number of common words between the two divided by the number of words in the longer of the two. This gives us a measure of word match that is comparable across instructions of different lengths.

b. TF-IDF: We use all the recipes in our training set to create a term frequency (TF)-inverse document frequency (IDF) vectorizer. Given an instruction from the evaluation set, we compute the TF-IDF vector for the instruction using this vectorizer. Given two instructions, we define their TF-IDF similarity as the cosine similarity between their TF-IDF vectors.

c. GloVe: We train GloVe embeddings (Pennington et al., 2014) on an in-domain corpus of 3 million words put together by combining text recipes and video transcriptions. Given an instruction, we average the GloVe embeddings (Pennington et al., 2014) of nouns and verbs[12] to obtain its embedding vector. Given two instructions, we define their embedding similarity as the cosine similarity of their embedding vectors.

d. BERT: Given an instruction, we compute its embedding vector using BERT-based sentence embedding (Reimers and Gurevych, 2019). We experiment with different variants and find that the BERT-base model trained on AllNLI, then on STS benchmark training set[13] performed the best for us. Given two instructions, we define their BERT similarity as the cosine similarity between their sentence embedding vectors.

e. RoBERTa: We also experiment with a variant of the above baseline where we use RoBERTa (Liu et al., 2019) instead of BERT to compute the sentence embeddings. We use RoBERTa-large trained on AllNLI, then on STS benchmark training set.

4.3 Model Ablations

We experiment with the following ablations of our unsupervised pairwise alignment model (§3.1):

HMM+IBM1 (nouns) We use the NLTK[14] part-of-speech tagger to identify all the nouns in an instruction and only use those to learn the IBM1 word-level alignments. This ablation is similar to the model proposed by Naim et al. (2014) that align objects in videos to nouns in text.

[12] We find that using only nouns and verbs outperforms using all words.

[13] https://pypi.org/project/sentence-transformers/

[14] https://www.nltk.org/

HMM+IBM1 (nouns and verbs) We use both nouns and verbs to learn IBM1 word-level alignments. This ablation is similar to the method used in Song et al. (2016) that align objects and actions in videos to nouns and verbs in text.

HMM+IBM1 (all words) We use all words (except stop words) in the *source* and the *target* recipe instructions to learn the word-level alignments.[15]

4.4 Evaluation Metrics

Given M source recipe instructions and N target recipe instructions, the alignment task is to label each of the M source instructions with a label from $[0, ..., (N-1)]$. Given a predicted sequence of labels (from baseline or proposed model) and a reference sequence of labels (from human annotations) for a recipe pair, we calculate the weighted-average[16] precision, recall and F1 score. We average these scores across all alignment pairs to compute aggregate scores on the test set.

4.5 Results

On text-video alignments Table 2 shows results of our pairwise alignment algorithm compared with baselines on 1,625 human aligned text-video recipe pairs from YouCook2. The BM25 baseline outperforms two of the textual similarity baselines. Within the textual similarity baselines, RoBERTa outperforms all others suggesting that a pretrained sentence level embedding acts as a good textual similarity method for this alignment task. The uniform alignment baseline, interestingly, outperforms all other baselines. This is mainly because in the YouCook2 dataset, the text instructions and the transcript sentences follow the same order, making uniform alignment a strong baseline. Our unsupervised HMM+IBM1 alignment model significantly outperforms (with $p < 0.001$) all baselines. Specifically, it gets much higher precision scores compared to all baselines. Under ablations of the HMM+IBM1 model, using all words to learn alignments works best.

On text-text alignments Table 3 shows results of our pairwise alignment algorithm compared with baselines on 200 human-aligned text-text recipe pairs from Common Crawl. Unlike text-video alignments, we find that the uniform alignment baseline does not outperform textual similarity baselines, suggesting that the different re-orderings between text-text recipe pairs makes alignment more challenging. Within textual similarity baselines, similar to text-video alignment, RoBERTa outperforms all others. We believe this is because text recipes tend to share similar vocabulary, making it easier to find similar words between two textual instructions. Video narrators tend to use more colloquial language than the authors of text recipes, making it more difficult to learn alignments using word similarities. Interestingly, both BM25 and RoBERTa get higher recall than our best HMM+IBM1 model but they lose out on precision. This suggests that retrieval models are good for identifying more alignments, albeit with lower precision. Our unsupervised HMM+IBM1 model again significantly outperforms ($p < 0.001$) all baselines on F1 score. Under ablations of the HMM+IBM1 model, we again find that using all words to learn alignments performs best.

Comparing text-video and text-text alignment results On comparing Table 2 and Table 3, we find that textual similarity baselines have overall higher scores on the text-text alignments than the text-video alignments. Our HMM+IBM1 model, on the other hand, has overall higher scores on text-video alignments than on text-text alignments. We attribute this contrast to the fact that two text recipes have higher vocabulary similarities than a text and a video recipe, resulting in textual similarity baselines to perform well on text-text alignments. Our HMM+IBM1 unsupervised learning model is able to do better on text-video pairs where the word usage differences are higher. Furthermore, the text-video pairs from YouCook2 are temporally aligned whereas the text-text pairs from Common Crawl have several re-orderings making the text-text evaluation set comparatively harder. The supplementary material includes an analysis of alignment outputs.

5 Data Release

We describe the data released in our MICROSOFT RESEARCH MULTIMODAL ALIGNED RECIPE CORPUS. In all our released data, for text recipes, we include the actual text of the instructions. Whereas, for video recipes, we release the URL to the YouTube video with timestamps corresponding to the aligned video segments.

[15] Experimental details of HMM+IBM1 model is in supplementary.

[16] Calculate metrics for each label, and find their average weighted by the number of true instances for each label.

Single Step	Multiple Steps
Beat eggs, oil vanilla and sugar together in a large bowl.	1.Beat eggs in large bowl until foamy. 2. Add sugar, oil and vanilla mix well.
Butter 2 loaf pans and bake 1 hour at 325 degrees.	1. Pour into greased muffin tins or loaf pans 2. Yields about 4 small loaves or 2 large. 3. Bake for 25 minutes.
Mix the zucchini, sugar, oil, yogurt and egg in a bowl.	1. Beat eggs, sugar, oil and vanilla. 2. Add zucchini.

Table 4: Three examples of single-step to multi-step breakdown from the pairwise alignments.

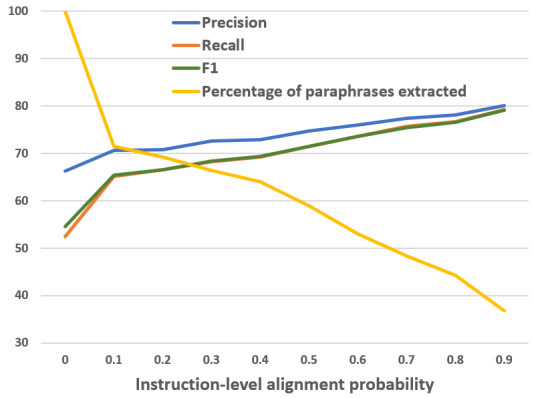

Figure 5: We plot the trade-off between the percentage of paraphrases extracted and the precision, recall and F1 score (as measured by human annotators) with increasing alignment probability threshold on 200 human-aligned text-text recipes pairs.

5.1 Pairwise and Joint Alignments

We release the pairwise alignments between recipes of the same dish (derived from § 3.1) for 4,262 dishes. This includes 63,528 alignments between text recipes, 65,432 alignments between text and video recipes; and 19,988 alignments between video recipes. We also release the joint alignments between multiple text and multiple video recipes within a dish (derived from §3.2) for 4,262 dishes.

5.2 Textual and Visual Paraphrases

The pairwise alignment algorithm described in §3.1 gives alignment probabilities for each pair of instructions it aligns. We threshold on these alignment probabilities to retrieve textual and visual paraphrases. Since our goal is to extract large number of high quality paraphrases, we decide on the threshold value by looking at the trade-off between the percentage of paraphrases extracted and their quality as measured by human annotators on 200 human-aligned text-text recipe pairs from our evaluation set (§4.1).

Figure 5 shows the trade-off between the precision, recall and F1 score and the percentage of paraphrases extracted with increasing threshold on instruction-level alignment probability. At 0.5 threshold, we extract 60% of the total alignments as paraphrases from our evaluation set. We use this threshold value of 0.5 on the pairwise alignments in the training, validation and test sets to extract a total of 358,516 textual paraphrases and 211,703 text-to-video paraphrases from 4,262 dishes and include it in our corpus.

5.3 Single-step to Multi-step breakdown

The pairwise alignments between text recipes include many instances where one instruction in one recipe is aligned to multiple instructions in another recipe with high alignment probability (greater than 0.9). Table 4 shows three such single-step to multi-step breakdown. We extract a total of 5,592 such instances from 1,662 dishes across the training, validation and test sets and include it in our corpus.

6 Applications of Our Corpus

We believe that our data release will help advance research at the intersection of language, vision and robotics. The pairwise alignment between recipes within a dish could be useful in training models that learn to rewrite recipes given ingredient or cooking method based constraints. The joint alignment over multiple text recipes within a dish should prove useful for learning the types of ingredient substitutions and instruction reordering that come naturally to expert cooks. The textual and visual paraphrases will, we believe, have implications for tasks like textual similarity, image and video captioning, dense video captioning and action recognition. The single-step to multi-step breakdown derived from our pairwise alignments may also prove useful for understanding task simplification, an important problem for agents performing complex actions.

Such multimodal data *at scale* is a crucial ingredient for robots to learn-from-demonstrations

of procedural tasks in a variety of environments. Collecting such large scale data is prohibitively expensive in robotics since it requires extensive instrumentation of many different environments. Other example applications are learning to ground natural language to physical objects in the environment, and catching when humans are about to commit critical errors in a complicated task and offering to help with corrective instructions.

7 Related Work

Alignment Algorithms Our unsupervised alignment algorithm is based on Naim et al. (2014), who propose a hierarchical alignment model using nouns and objects to align text instructions to videos. Song et al. (2016) further build on this work to make use of action codewords and verbs. Bojanowski et al. (2015) view the alignment task as a temporal assignment problem and solve it using an efficient conditional gradient algorithm. Malmaud et al. (2015) use an HMM-based method to align recipe instructions to cooking video transcriptions that follow the same order. Our work contrasts with these works in two ways: we learn alignments between instructions that do not necessarily follow the same order; and our algorithm is trained on a much larger scale dataset.

Multi-modal Instructional Datasets Marin et al. (2019) introduce a corpus of 1 million cooking recipes paired with 13 million food images for the task of retrieving a recipe given an image. YouCook2 dataset (Zhou et al., 2018a) consists of 2,000 recipe videos with human written descriptions for each video segment. The How2 dataset (Sanabria et al., 2018) consists of 79,114 instructional videos with English subtitles and crowdsourced Portuguese translations. The COIN dataset (Tang et al., 2019) consists of 11,827 videos of 180 tasks in 12 daily life domains. YouMakeup (Wang et al., 2019) consists of 2,800 YouTube videos, annotated with natural language descriptions for instructional steps, grounded in temporal video range and spatial facial areas.

Leveraging Document Level Alignments Our work relies on the assumption that text recipes and instructional cooking videos of the same dish are comparable. This idea has been used to extract parallel sentences from comparable corpora to increase the number of training examples for machine translation (Munteanu and Marcu, 2005; Abdul-Rauf and Schwenk, 2009; Smith et al., 2010; Grégoire and Langlais, 2018). Likewise, TalkSumm (Lev et al., 2019) use the transcripts of scientific conference talks to automatically extract summaries. Zhu et al. (2015) use books and movie adaptations of the books to extract descriptive explanations of movie scenes.

Related Tasks A related task is localizing and classifying steps in instructional videos (Alayrac et al., 2016; Zhukov et al., 2019) where they detect when an action is performed in the video whereas we focus on describing actions. Dense event captioning of instructional videos (Zhou et al., 2018b; Li et al., 2018; Hessel et al., 2019) relies on human curated, densely labeled datasets whereas we extract descriptions of videos automatically through our alignments.

8 Conclusion

We introduce a novel two-stage unsupervised algorithm for aligning multiple text and multiple video recipes. We use an existing algorithm to first learn pairwise alignments and then use a graph-based algorithm to derive the joint alignments across multiple recipes describing the same dish. We release a large-scale dataset constructed using this algorithm consisting of joint alignments between multiple text and video recipes along with useful commonsense information such as textual and visual paraphrases; and single-step to multi-step breakdown.

Although our dataset focuses on the cooking domain, our framework should generalize to any domain with abundant volumes of unstructured-but-alignable multi-modal data. DIY (Do-It-Yourself) videos and websites, for instance, are an obvious next target. We also envision extending this work by including audio and video features to enhance the quality of our alignment algorithm. Ultimately, we believe this work will further the goal of building agents that can work with human collaborators to carry out complex tasks in the real world.

Acknowledgments

We would like to thank Harpreet Sawhney, Roshan Rao, Prasoon Goyal, Dilip Arumugam and Raymond J. Mooney for all their help. We would also like to thank the four anonymous reviewers for their useful comments and suggestions.

References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece. Association for Computational Linguistics.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. 2016. Unsupervised Learning from Narrated Instruction Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583.

Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. 2015. Weakly-Supervised Alignment of Video with Text. In *Proceedings of the IEEE international conference on computer vision*, pages 4462–4470.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

DeepMind. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. *https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii*.

Francis Grégoire and Philippe Langlais. 2018. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. 2019. A case study on combining ASR and visual features for generating instructional video captions. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 419–429, Hong Kong, China. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2125–2131, Florence, Italy. Association for Computational Linguistics.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly Localizing and Describing Events for Dense Video Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7492–7500.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nicholas Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 143–152, Denver, Colorado. Association for Computational Linguistics.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Iftekhar Naim, Young Chol Song, Qiguang Liu, Henry Kautz, Jiebo Luo, and Daniel Gildea. 2014. Unsupervised Alignment of Natural Language Instructions with Video Segments. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

OpenAI. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*

and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A Large-scale Dataset for Multimodal Language Understanding. In *NeurIPS*.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. 2019. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model.

Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Unsupervised Semantic Parsing of Video Collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4480–4488.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411, Los Angeles, California. Association for Computational Linguistics.

Young Chol Song, Iftekhar Naim, Abdullah Al Mamun, Kaustubh Kulkarni, Parag Singla, Jiebo Luo, Daniel Gildea, and Henry A Kautz. 2016. Unsupervised Alignment of Actions in Video with Text Descriptions. In *IJCAI*, pages 2025–2031.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1207–1216.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Weiying Wang, Yongcheng Wang, Shizhe Chen, and Qin Jin. 2019. YouMakeup: A large-scale domain-specific multimodal dataset for fine-grained semantic comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5133–5143, Hong Kong, China. Association for Computational Linguistics.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards Automatic Learning of Procedures from Web Instructional Videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7590–7598.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-End Dense Video Captioning with Masked Transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. 2019. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3545.

A Supplemental Material

In this supplementary, we describe the details of our data collection process (§A.1), experimental details of our algorithm (§A.2) and provide analysis of our alignment outputs (§A.3).

A.1 Details of Data Collection

A.1.1 Common Crawl Text Recipes

We use recipe data from Common Crawl [17] that has metadata formatted according to the Schema.org Recipe schema [18] including title, ingredients, instructions, and a URL to the recipe source. There were originally 3.2 million recipes extracted from Common Crawl. We filter the data by limiting the data to recipes with instructions written in English, removing recipes with titles that are longer than 5 words, removing duplicate recipes, removing recipes where the recipe title contains words that are not in the top 50% most common words

[17] https://commoncrawl.org/

[18] https://schema.org/Recipe

that occur in the recipe titles, and removing recipes with fewer than 2 steps. After filtering the data, we clustered the recipes into dishes using exact match on the recipe titles. We only retain recipes from dishes that have at least three recipes. The final dataset has a total of 4,262 dishes and 48,852 recipes with an average of 8 instructions per recipe.

A.1.2 YouTube Video Recipes

Given the dish names from the text recipes, we extract YouTube video recipes for each of the dishes. The number of videos extracted for each dish is proportional to the number of text recipes found for that dish. For instance, for a more popular dish like *chocolate chip cookies*, we would extract more text and video recipes than for a less popular dish like *creme brulee*. The number of videos extracted ranges from 3 to 100.

A.1.3 Chat/Content Classifier

Instructional cooking videos can contain a lot of non-instructional content ("chat"). For example, the person cooking the dish often introduces themselves (or their video channel) at the beginning of the video. They sometimes also introduce the dish they are going to prepare and suggest pairings for the dish. The non-instruction content are often found in the beginning and towards the end of the video but there are several instances of "chat" interspersed with instructional content as well. Since we wish to align these videos to text recipe instructions that do not contain non-instructional information, we need a way to remove non-instructional content. We train a supervised neural network based classifier for this task.

We train our classifier using the YouCook2 dataset (Zhou et al., 2018a) of 1,500 videos across 90 dishes. This dataset was created by asking humans to identify segments of a video that correspond to an instruction and annotate each segment with an imperative statement describing the action being executed in the video segment. We make the assumption that the transcript sentences that are included within an annotated video segment are instructional whereas those that are not included within an annotated video segment are non-instructional. We first transcribe all 1,500 videos in the dataset using a commercial transcription web service. We split the transcription into sentences using a sentence tokenizer. We label a transcript sentence with the label 1 if the corresponding video segment was annotated and with the label 0 if it was not. We get a total of 90,927 labelled transcript sentences which we split by dishes into the training (73,728 examples), validation (7,767 examples) and test (9,432 examples) sets.

We use an LSTM (long-short term memory) model (Hochreiter and Schmidhuber, 1997) with attention (Luong et al., 2015) to train a binary classifier on this data. We initialize (and freeze) our 300-dimensional word embeddings using GloVe (Pennington et al., 2014) vectors trained on 330 million tokens that we obtain by combining all text recipes and transcript sentences. We use the validation set to tune hyperparametrs of our LSTM classifier (hidden size: 64, learning rate: 0.00001, batch size: 64, number of layers: 1). Our chat/content classifier achieves 86.76 precision, 84.26 recall and 85.01 F1 score on the held out test set.

A.1.4 Recipe Pair Pruning Strategy

We define the following two pruning strategies to reduce the number of extracted recipe pairs:

Ingredient match: Each of our text recipes from Common Crawl contains an ingredients list. Video recipes from YouTube however do not contain ingredient lists. We therefore estimate the ingredients for video recipes using text recipes of the same dish. We construct a set of ingredients at the dish level by combining all ingredients of the text recipes within that dish. We then use this dish-level ingredients information to identify ingredient words from the words of video transcriptions. Given a recipe pair, we compare the ingredients of the two recipes and if the percentage of ingredients that match is below a threshold, we remove the pair. For text-text and text-video recipe pair, we set this threshold to be 70%, whereas for video-video recipe pair, we set this threshold to be 90% (since video-video recipe pairs tend to be more noisy).

Instruction length match: For text-text recipe pairs, if number of instructions in one recipe is more than double the number of instructions in another recipe, we remove the pair. For video recipes, if there are more than 100 sentences in the transcript after removing the background sentences, we remove that video recipe.

A.2 Details of HMM+IBM1 Model

We train the HMM+IBM1 pairwise alignment model on three kinds of recipe pairs: text-text, text-video and video-video. The lower level IBM1 model works on words of text instruction or transcript sentences. The vocabulary size of all the

text recipes from 4,262 dishes put together totals to 48,609 words. Since most words do not appear very frequently across the text recipes corpus, we reduce the vocabulary size to 13,061 by removing words that occur fewer than 5 times in the training set. Likewise, we reduce the vocabulary size of video recipe transcriptions to 16,733 words (from 88,744 words) by removing words that occur fewer than 15 times in the training set. We first train the HMM+IBM1 model for 3 iterations with a jump range of $[-1, 0, +1]$ and further train it for 2 iteration with a jump range of $[-2, 0, +2]$. We find that warm starting the model with a shorter range helps the model to learn better alignments.

A.3 Alignment Output Analysis

Table 5 shows the alignment between two text recipes for *chocolate chip cookies* obtained by our pairwise algorithm. The alignment task here is to align each instruction in the source recipe to one of the instructions in the target recipe. The table displays all the instructions in the source recipe in the second column. The first column of the table displays instructions from the target recipe that aligns to the source recipe instruction in the same row. The sentence level probabilities are shown in the last column.

We can see the reordering between the two recipes by comparing the instruction indices. We see that instructions 0 to 2 from the source are aligned to target instructions with very high probabilities suggesting they are close paraphrases. Instruction 3 and 8 from the source, on the other hand, are aligned with comparatively lower probabilities to the target and we can see that in these two cases, the two instructions do differ in meaning. Instructions 6,7 and 8 (in source) aligned to instruction 11 (in target) is an example of single step to multi-step breakdown.

Target recipe instruction	Source recipe instruction	Probability
0: Preheat your oven to 350 degrees F.	0: Preheat the oven to 350 degrees F.	0.9999
2: In the bowl of your mixer cream together your butter and sugars until light and fluffy about 3-5 minutes.	1: In a large bowl or the bowl of a stand mixer cream the butter sugar brown sugar eggs & vanilla together until smooth & fluffy.	0.9998
1: Sift together the flour baking soda baking powder and salt into a medium sized bowl and set aside.	2: In another bowl whisk together the flour salt baking powder and baking soda.	0.9997
4: Add in the vanilla and mix.	3: Add this to the butter mixture and mix until well combined.	0.6889
6: Fold in your chocolate until evenly added throughout the dough.	4: Stir in the chocolate chips.	0.9820
8: Scoop your dough out onto the sheets.	5: Form the dough into golf-ball sized balls and place them about 2 inches apart on a baking sheet.	0.9997
11: Bake 10-12 minutes for smaller cookies or 18-20 minutes for larger cookies.	6: Bake for 9-10 minutes just until the edges start to brown lightly.	0.9912
11: Bake 10-12 minutes for smaller cookies or 18-20 minutes for larger cookies.	7: Do not overbake them or they will be crispy rather than chewy.	0.9528
11: Bake 10-12 minutes for smaller cookies or 18-20 minutes for larger cookies.	8: They still look underbaked when you take them out but will firm up as they cool.	0.6465
12: Allow the cookies to cool slightly on your baking sheet then move them to another surface to cool completely.	9: Let them cool on the pan for about 5 minutes and them move to a wire rack to cool completely.	0.9973
14: Store in an air-tight container at room temperature for up to 3 days or freeze for up to 2 months.	10: Cookies will keep for 7 days in a sealed container at room temperature.	0.8309

Table 5: Alignment between two text recipes of *chocolate chip cookie* with their sentence level probabilities.