Propositional Extraction from Natural Speech in Small Group Collaborative Tasks

Videep Venkatesha
Colorado State University

Abhijnan Nath
Colorado State University

Ibrahim Khebour
Colorado State University

Avyakta Chelle
Colorado State University

Mariah Bradford
Colorado State University

Jingxuan Tu
Brandeis University

James Pustejovsky
Brandeis University

Nathaniel Blanchard
Colorado State University

Nikhil Krishnaswamy
Colorado State University

{videep.venkatesha, nkrishna}@colostate.edu

ABSTRACT

In the realm of collaborative learning, extracting the beliefs shared within a group is paramount, especially when navigating complex tasks. Inherent in this problem is the fact that in naturalistic collaborative discourse, the same propositions may be expressed in radically different ways. This difficulty is exacerbated when speech overlaps and other communicative modalities are used, as would be the case in a co-situated collaborative task. In this paper, we conduct a comparative methodological analysis of extraction techniques for task-relevant propositions from natural speech dialogues in a challenging shared task setting where participants collaboratively determine the weights of five blocks using only a scale. We encode utterances and candidate propositions through language models and compare a cross-encoder method, adapted from coreference research, to a vector similarity baseline. We see substantially increased performance when using the cross-encoder and establish a novel baseline on this challenging task. Further, we extend our examination to transcripts generated by Google's Automatic Speech Recognition system, to assess the potential for automating the propositional extraction process in real-time. This study not only demonstrates the feasibility of detecting collaboration-relevant content in unstructured interactions but also lays the groundwork for employing AI to enhance collaborative problem-solving in classrooms, and other collaborative settings, such as the workforce. Our code may be found at: `https://github.com/csu-signal/PropositionExtraction`

Keywords

Collaborative Problem Solving, Propositional Extraction, Natural Speech

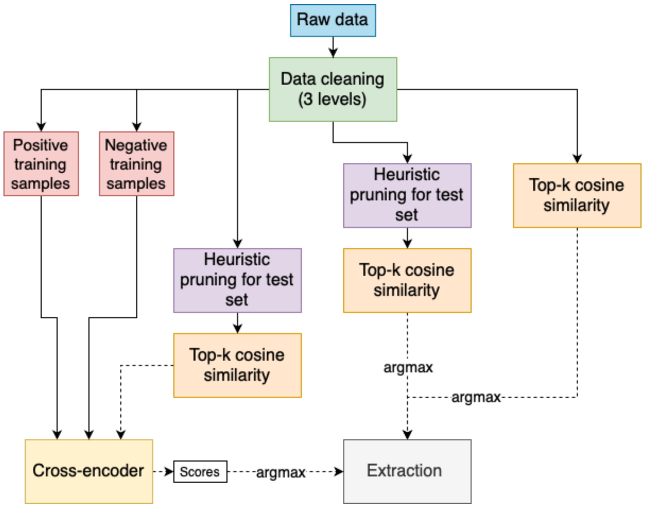

Figure 1: Schematic overview of propositional extraction approach. Dashed lines indicate inference samples.

1. INTRODUCTION

For computer-assisted education, an important capability of automated systems is the ability to extract the meaning from student sentences or utterances to determine what they know, infer, or understand in the course of a task, activity, or assignment. In a naturalistic situated dialogue, like a small group in a classroom, information exchange is likely to consist of overlapping utterances with references grounded in the situational context, such as to objects in the scene or actions taken. Therefore, unlike in idealized scenarios such as strict turn-taking dialogues or written texts, it may be difficult to determine the exact semantic or propositional content that is being expressed by a single utterance.

An added challenge for educationally-grounded AI tasks such as knowledge tracing [33] is that the same semantics or proposition may be expressed in natural speech in radically different ways—there are likely to be incomplete sentences, repetition or restatement, filler words or disfluencies—and extracting relevant meaning despite such noise is crucial if an automated system is to make correct inferences about what students know or understand about their activity.

The propositional content that students assert is critical to tracking the collaborative process as students share their understanding and build consensus or common ground [36, 24]. For example, an automated agent for collaborative problem solving support would need to track surfaced propositions as a measure of task progression. Additionally, students in collaborative settings achieve greater learning outcomes when they engage in *leading* the discussion, which involves making new claims and not simply reiterating previously-stated information [40]. The ability to extract propositional content from dialogue provides a way for an agent to determine whether a claim was already stated within the group. This would provide a necessary feature to determine whether a student is helping to lead the task forward, thereby enabling better prediction of learning outcomes from mined data.

In this paper, we take the transcribed utterances of a shared collaborative task, which are annotated with ground truth task-relevant propositions that are expressed therein, and use cosine similarity and cross-encoder methods to extract the propositions from the utterance text. Fig. 1 shows a schematic overview of our approach. We also extend our methods to utterances automatically segmented and transcribed by Google Cloud Platform's Automated Speech Recognition, showing how our propositional extraction methods may be incorporated into an automated system with a relatively low level of degradation due to automated transcription. Our results show the utility of methods adapted from coreference research in the field of natural language processing on this challenging task.

Our novel contributions are:

- Establishment of a novel, challenging task of propositional extraction from natural speech during a collaborative interaction.
- Comparison of cosine similarity and cross-encoder methods using multiple language models and levels of data cleaning, establishing a novel baseline and theoretical upper bound on the best performance of our methods in this task.
- Assessment of the level of performance degradation introduced by automated speech transcription when compared to manually transcribed utterances.

Our code may be found at `https://github.com/csu-signal/PropositionExtraction`.

2. BACKGROUND AND RELATED WORK

Collaborative tasks concern the construction and maintenance of a shared conception of the problem at hand [35], involving mutual engagement and coordinated effort to solve the problem together. Within such a framework, especially one centered around shared synchronous tasks, quantity of specific propositions discussed has been shown to be a significant predictor of learning gains [12]. Therefore, propositional extraction serves an important role in automated analysis of shared task data in an educational context, or for an automated system to make inferences about construction of shared knowledge in real time.

Propositional extraction. Prior work on propositional extraction from natural language has primarily been conducted from written texts in domains such as question answering, where early methods relied on approaches such as semantic memory [10]. Classical machine learning approaches like support vector machines have been applied to opinion mining to find "propositional opinions," or sentence fragments that contain the object of an assertion, incorporating word and feature-level knowledge from resources like WordNet, FrameNet, and PropBank [4]. Linguistic features have even been used to extract "ideas" from transcribed speech in the clinical domain, as a technique to predict Alzheimer's disease and other types of cognitive decline [8]. These early works not only show the utility of propositional extraction in various domains, but also demonstrate the relative sparsity of study on this topic. With the advent of neural network methods for text processing, these have been applied to NLP problems like propositional extraction from argumentation and rhetoric [20, 21]. These approaches include reported speech, as may appear in documents such as news articles. To the best of our knowledge, we are the first to attempt a similar task on transcribed naturalistic speech data from a collaborative task setting reminiscent of small group work in classrooms.

Pairwise Representation Learning. All of the aforementioned approaches frame the problem as one of establishing a mutual relationship between a piece of text from a dataset and another piece of text from a library of candidates, be they ideas, opinions, or propositional information more generally. Pairwise representational learning techniques have long been popular in the deep learning community for learning such relationships between two pieces of text. While some previous works modeled these relationships for text-generation tasks like abstractive document summarization [29], machine-comprehension [17], or document-reconstruction [26], others have also explored pairwise learning to compute similarity metrics between pairs of documents [2, 34, 43] as well as for masked language modeling [11]. More recently, for clustering-related tasks like coreference resolution, a "cross-encoding" framework has been used to learn pairwise features of possible coreferent mentions [1, 5, 6, 16, 41, 42]. These works, originally inspired by [18], learn high-level semantic features of a mention (e.g., of an entity or event) within a sentence in the context of another mention-containing sentence and compute the coreference probabilities of such pairs before clustering mentions that refer to the same entity. We adopt this "cross-encoding" technique for both our candidate proposition generation procedure, as well as for calculating the probability of a given utterance referring to a candidate proposition.

Cross-Encoders. According to discourse coherence theory, in a dialogue between two or more participants, the content of the discussion is essentially a subset of the common knowledge, beliefs, and common intention (goal) that each participant has at any given point. As such, certain processing decisions like identifying referring expressions or detecting common propositional content between utterances can be made locally within the "attentional state" of the discourse [14, 15]. For instance, in a collaborative problem-solving setting, the words in an utterance that any participant uses to describe a specific sub-task within the overall task, are constrained by "discourse segment purpose" or their common intention at that specific point in the dialogue. This constraint in the appearance of utterances to maintain coherence in the collaborative problem-solving dialogue allows us to map an utterance to a proposition by focusing only on the *local* elements in the utterance/proposition pairs.

However, since linguistic constraints or rule-based heuristics used to determine this attentional state can be narrow in their scope or domain-specific, most previous works have modeled the attentional state using neural networks [7, 16, 19]. These models are typically built on top of pretrained transformer-based language models (LMs) [39] like RoBERTa or Longformer [3, 28] that are known to capture rich semantic features through their contextualized representations of tokens and sequences. Apart from computationally modeling the innate structural coherence in a discourse, these architectures can also generate potential referents by demarcating the attentional state within a dialogue, through context.

These works have focused on various natural language understanding (NLU) tasks, including coreference resolution. Our task is adjacent to coreference resolution since we have to map a set of utterances to their corresponding propositions in a collaborative dialogue. As such, we take inspiration from the pairwise scorer/cross-encoder architecture commonly used as a pairwise representation learning framework in cross-document coreference resolution (CDCR) [1, 5, 6, 30, 31, 41, 42]. In this technique, a classifier is forced to learn a combined representation of one mention (represented by a trigger word) in the context of the other, both of which are encoded within their respective sentences. This learning strategy is an effective way to generate similarity scores between pairs of event or entity mentions due to the contextualized learning framework.

3. DATASET

The Weights Task [23] is a situated collaborative problem-solving (CPS) task wherein groups of 3 work together to deduce the weights of differently colored blocks using a balance scale. There are a total of 10 groups, resulting in approximately 4 hours of audiovisual data. Participants consented to the release of their likenesses for research purposes. The study protocol and release of A/V data were approved by the Colorado State University institutional review board.[1] In this work we focus on Phase 1 of the task, where the group has five blocks of different colors ($C = \{\text{red, yellow, green, blue, purple}\}$) whose weights follow an instance of the Fibonacci sequence ($W_n = \{10g, 10g, 20g, 30g, 50g\}$). At the start of the task, the group is told that the red block weighs 10 grams.[2]

For our purposes, the Weights Task Dataset (WTD) contains speech transcribed manually by humans (hereafter referred to as "Oracle" transcriptions) as well as speech transcribed automatically by Google Cloud Platform's Automatic Speech Recognizer (Google ASR). The Oracle and Google transcription processes also *segmented* the speech into utterances—a single person's continuous speech, delimited by silence. There are a total of 2,140 utterances that contain transcribed speech according to Oracle segmentation, and 1,500 utterances containing transcribed speech according to Google segmentation.

Due to the overlapping nature of speech in this setting, utterance segmentation leads to many sentence fragments and overlaps, as well as mistranscription by the automated system, which leads to challenges in extracting the intended meaning behind any given utterance. An additional challenge to meaningful information extraction from the linguistic channel is that due to the multimodal nature of the task, a complete interpretation of an utterance may require recourse to another modality. For example, someone may say "this one" while pointing to a specific block. The pointing makes it clear which block is being referred to but without access to the video showing where the person is pointing, the language alone is ambiguous. The above factors enumerate the challenges to extracting propositions expressed through dialogue in this setting.

The propositions themselves are annotated in the context of the *common ground* that evolves between group members as the task proceeds, that is, the set of propositions Φ each individual comes to believe as factual and that the group must agree upon, implicitly or explicitly, to arrive at the goal [32]. In the case of the Weights Task, the participants must all arrive at the correct assignments of weight $w \in W$ to color $c \in C$ to solve the task. The WTD is annotated with the propositions that are asserted, evidenced, or agreed upon as the task unfolds, based upon the multiple modal channels and prior context. Our goal is to recover those propositions from the transcribed speech.

3.1 Preprocessing and Annotation

From the Phase 1 data of the WTD we removed utterances spoken by the study researcher as she introduced the task and setup, to focus only on dialogues within the group. Because of the multimodal nature of the task and the prevalent use of demonstratives, we enriched the transcribed utterances using a "dense paraphrasing" method inspired by Tu et al. [37, 38], that rewrites a textual expression to reduce ambiguity and make explicit the underlying semantics. We isolated the utterances containing at least one pronoun from a predefined set, performed a partial assignment of blocks referenced by those pronouns based on actions that overlapped the utterances, and had annotators identify the blocks denoted by the remaining pronouns, if any, while referring to the video (see Fig. 2). This annotation was performed separately for the Oracle and Google transcription. Utterances were dually annotated, resulting in an average Cohen's $\kappa = 0.89$ over the Oracle transcriptions and $\kappa = 0.87$ over the Google transcriptions, indicating high annotator agreement [9]. A gold standard was then generated through adjudication by an expert. The original utterances were then replaced with the dense paraphrased versions. High agreement scores and accuracy metrics demonstrate the reliability and effectiveness of the annotation process. This procedure *decontextualizes* the utterances from their multimodal de-

[1] The dataset and consent documents associated with the original study protocol are publicly available at https://zenodo.org/records/10252341.

[2] Although a gram is a unit of mass, the colloquial dialogue in the dataset uses "mass" and "weight" interchangeably.

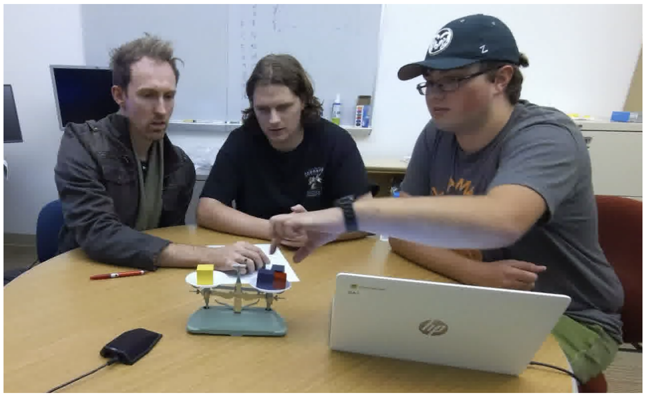

Figure 2: Example of dense paraphrasing with reference to video. The original utterance is "we can replace one of [these] with the twenty." With reference to the video, an annotator can see the rightmost participant reaching for the red and blue blocks, so the dense paraphrased utterance is "we can replace one of *red block, blue block* with the twenty."

pendencies, allowing us to evaluate the utterance as though it were text only.

4. METHODS

We investigated two methods for extracting propositional content from utterances: a *cosine similarity* baseline, and a *cross-encoder* adapted from entity and event coreference research in the field of natural language processing (NLP). These were both evaluated over the Oracle transcriptions of utterances, and the Google automatic transcriptions, and using various levels of data cleaning to explore performance of the different methods in settings that range from more idealized to more realistic. Below we describe the methodology for cleaning the data and training the cross-encoder.

Propositional content in the Weights Tasks takes the form of a relation between a block and a weight value (e.g., $red = 10$), between two blocks (e.g., $red = blue$), or between one block and a combination of other blocks (e.g., $red < blue + green$). To generate all possible candidate propositions in the domain, we employed a systematic process that combined the 5 block colors (red, blue, green, purple, yellow), 5 potential weights (10, 20, 30, 40, and 50), and 4 relations ($=$, $\neq$, $<$, $>$) into all possible combinations that fit the aforementioned formats. "Conjunctive" propositions (e.g., $green > 20$ *and* $yellow < 50$) were also allowed, up to a length of 3 conjuncts (the maximum that ever appeared in the actual dataset). We normalized all candidate propositions for the symmetric property of equality (e.g., so that $red = blue$ is the same as $blue = red$), and dropped the resulting duplicates. The result was 5,005 total candidate propositions that *could* be expressed in the Weights Task domain.

Any given proposition might be expressed in multiple ways. For instance, in the data "purple block's thirty," "purple one thirty," "let's go thirty purple block's thirty," and "teeter teeter purple block's less forty greater twenty purple block's likely thirty" all appear as ways of expressing the proposition $purple = 30$, despite the fact that they may contain extra words or even mentions of additional blocks or weights not contained within the proposition actually expressed. We therefore modeled propositional extraction as a type of *coreference* problem, where the goal is not to determine whether two entity mentions refer to the same thing [25], but rather to determine if two utterances mention both the same entity (block) and the same property (weight or relation).

4.1 Data Cleaning

Filtration of the dataset is motivated by the fact that many utterances, even after dense paraphrasing, still do not mention a specific object or weight, meaning that extracting an object-weight or object-object relation from the utterance alone is infeasible. Our filtration steps follow steps used in existing coreference research [1]. The decision to follow this methodology was made at the outset before any experimental results were available. We adopted three levels of data cleaning. 1) The first level of cleaning consisted of removing all instances where neither color nor weight was mentioned in the transcript. An example of an utterance removed at this step would be "i mean it's not gonna go anywhere i guess it's just oh." 2) The second level of cleaning involved removing all utterances where the mentioned colors and weights did not match the annotated proposition. For example, in an utterance "yeah red block, blue block should be twenty as well", "yeah" is actually an acceptance of a previously asserted proposition (in this case $green = 20$), and $red + blue = 20$, the mention of which is in the utterance, is not a valid propositional form in the task domain as the left hand side must be a single block (in this case, the truth of $red + blue = 20$ is implicit in two other (valid) propositions $red = 10$ and $blue = 10$). 3) The final level of cleaning removed all instances that do not mention a color, but only a weight. For instance, the utterance "well the top is a ten" is annotated as $blue = 10$, but with only the text, even a human would struggle to identify the correct proposition. The dataset annotators, meanwhile, had access to the video and could see that the top block referred to is blue, but as we focus only on transcriptions of natural speech, this information is not available to our method.

We encoded utterances as vector representations in 3 language models: **BERT**`-base-uncased` [11], **RoBERTa**`-base` [28], and **Longformer**`-base-4096` [3]. Before encoding, stop words were filtered out according to a standard list augmented with words that occurred in 5 or fewer bigrams over all the transcriptions, and are not number words, color words, or (in)equality relation words. To retrieve the vectors, we summed over the last 4 encoder layers of each model and took the average of the `[CLS]`/`<bos>` token vector and all individual token vectors in the utterance. These vectors were used for propositional extraction by comparison using cosine similarity, and for training the cross-encoder architecture.

4.2 Cross-Encoder

Above, and in Sec. 2, we motivated propositional extraction as a type of coreference problem. Therefore, we use a cross-encoder neural network that is common in NLP approaches to coreference. The cross-encoder learns a paired "contextualized" representation for an utterance proposition pair. Unlike previous coreference approaches mentioned in Sec. 2, which focus on the specific trigger word within a sentence, we encode the *entire* utterance in the context of

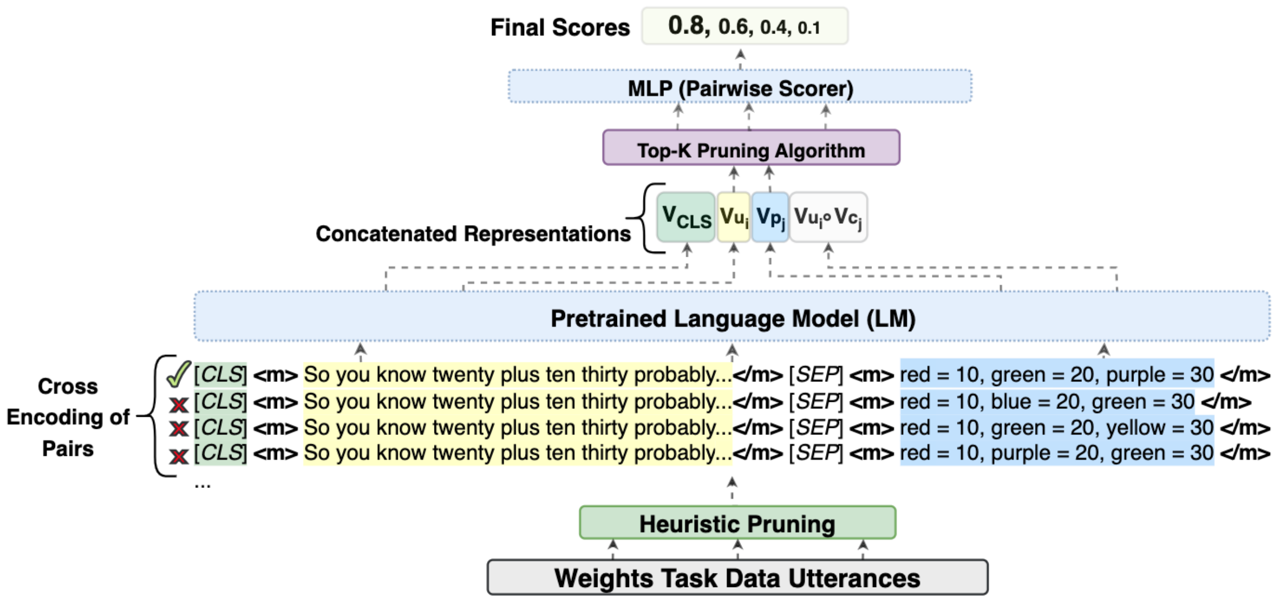
Final Scores
0.8, 0.6, 0.4, 0.1
MLP (Pairwise Scorer)
Top-K Pruning Algorithm
Concatenated Representations
V_CLS
Vu_i
Vp_j
Vu_i∘Vc_j
Pretrained Language Model (LM)
Cross Encoding of Pairs
[CLS] <m> So you know twenty plus ten thirty probably...</m> [SEP] <m> red = 10, green = 20, purple = 30 </m>
[CLS] <m> So you know twenty plus ten thirty probably...</m> [SEP] <m> red = 10, blue = 20, green = 30 </m>
[CLS] <m> So you know twenty plus ten thirty probably...</m> [SEP] <m> red = 10, green = 20, yellow = 30 </m>
[CLS] <m> So you know twenty plus ten thirty probably...</m> [SEP] <m> red = 10, purple = 20, green = 30 </m>
...
Heuristic Pruning
Weights Task Data Utterances

Figure 3: Schematic overview of the cross-encoder architecture.

the proposition to generate a combined representation for an utterance/proposition pair. This is for two reasons: firstly, in our framework, both the transcript and the candidate proposition can contain more than one color mention, which serves as a trigger indicating a block. For instance, consider "so *purple* block, *blue* block should be forty right there" (utterance) and $purple + blue = 40$ (candidate proposition). Encoding the utterance once for each specific color-trigger using a language model could drastically increase computational cost without any additional benefits of contextualization. This could also likely break down higher-level semantic signals that can otherwise be encoded with a wider context-window or the entire sentence. Secondly, under certain lenient pruning strategies, some transcripts may not contain any color at all. E.g., "... so you know twenty plus ten thirty probably ..." with a candidate proposition $red = 10 \wedge green = 20 \wedge purple = 30$. In such cases, full sentential context may capture more subtle semantic signals that are crucial for this task.

For an utterance/proposition pair (u_i, p_j), we construct an overall representation of the pair using the language model encoder. This representation consists of four individual parts, following modern standard practice in coreference established by Caciularu et al. [5]. We first surround u_i and p_j individually with special tokens `<m>` and `</m>` that are added to the language model tokenizer vocabulary and acquire learned representations during the training process. The first part of this overall representation is V_{CLS}, the pooled representation (`[CLS]`/`<bos>`) token of the last encoder hidden state). This representation is often used as a classification token in NLP tasks. Then, we encode u_i and p_j individually in the *context* of each other (that is, u_i when preceding p_j and p_j when following u_i)[3]. These comprise the second and the third components of the overall representation: V_{u_i} and V_{p_j}. We then encode the element-wise, or Hadamard product of these two representations ($V_{u_i} \odot V_{p_j}$) to provide further cross-attention based signals. These four individual representations are then concatenated into a unified representation ($[V_{CLS}, V_{u_i}, V_{p_j}, V_{u_i} \odot V_{p_j}]$), which is fed into a multi-layer perceptron (MLP) to get similarity scores between the utterance and proposition (Eq. 1) The MLP is a two-layer neural network (768 and 128 neurons) that takes in the concatenated representation (768×4 = 3072 dimensions) and outputs a scalar, or after a sigmoid operation, the probability of an utterance referring to a proposition.

$$Score(u_i, p_j) = MLP([V_{CLS}, V_{u_i}, V_{p_j}, V_{u_i} \odot V_{p_j}]) \qquad (1)$$

The candidate proposition with the highest score is retrieved, or the scores can be used to compute a *ranking* of candidate propositions, for metrics like top-k accuracy. Fig. 3 shows a schematic overview of the cross-encoder architecture.

4.2.1 Cross-Encoder Training

The parameters of the MLP are learned along with the parameters of the pretrained language model. Motivated by [1], we use a symmetric cross-encoding framework that minimizes the mean of the Binary Cross Entropy (BCE). More specifically, an utterance (u_i) and a proposition (p_j) are encoded bidirectionally, by interchanging their sequential positions in the input text ((u_i, p_j) and (p_j, u_i)). This results in a different unified representation in each direction and we minimize the average of the BCE loss over the encodings in both directions. Mathematically,

$$\mathcal{L}_{\text{BCE}(\theta,\,\phi)} = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log\left(1 - \hat{y}_i\right)\right) \qquad (2)$$

[3] The positional encoder of transformer models cause the resulting representations to be different despite the input order being the same.

where y and $\hat{y}$ are the true and predicted probabilities for an utterance-proposition encoding in one of the directions in a sample batch of size m. θ and ϕ are the parameters of the MLP and the pretrained LM, respectively. We train using a batch size of 20 for 12 epochs, with a learning rate of $1e-6$ on the LM parameters and $1e-4$ on the MLP pairwise scorer.

4.3 Experiments

Cross-Encoder. As mentioned, our data suffers from an imbalance between negative and positive samples, in that the vast majority of candidate propositions are not matches for a given utterance. This phenomenon is also present in common event coreference datasets, which results in a training dataset that is severely imbalanced toward negative pairs if not handled [1]. In our case, it is usually quite obvious when a candidate proposition is not a possible match for an utterance because the candidate does not contain the object or weight value mentioned in the utterance. Therefore, we employ a heuristic pruning strategy that operates at 2 levels. 1) we compare all propositions that include both the color and weight mentioned in the utterance (e.g., candidate matches for an utterance containing "red" and "ten" would include $red = 10$, $red \neq 10$, $red < 10$, etc.) 2) If the list of candidates is still empty, as might be the case if the utterance is simply "it's fifty!", we then enlarge the search space by getting all the propositions that contain any of the colors or weights mentioned in the utterance. This process is similar to the lemma-based heuristic pruning used for training a cross-encoder for cross-document event coreference by Ahmed et al. [1].

After filtering the candidate propositions with heuristic pruning, to create the training dataset for the cross-encoder, we pair an utterance with its annotated correct proposition as a positive pair and choose 4 random propositions from the filtered candidate propositions and pair them with the utterance as negative pairs. For example, the utterance "ok so the red has ten" would be a positive match with $red = 10$ and a negative match with only 3 other candidates generated after pruning. This results in a more balanced ratio of negative to positive candidate propositions for a given utterance, which is beneficial for training. The random selection from the filtered propositions ensures a diverse and robust set of negative samples. We pick only 4 random negative samples because a significant number of annotated propositions are of the form <color, relation, weight> which means that after the first level of heuristic pruning, certain transcripts would have only 4 possible candidate propositions, viz. $<color>$ $\{=, \neq, <, >\}$ $<weight>$.

We perform a rotating leave-one-group-out experiment where cross-encoder training is performed over 9 of 10 groups in the WTD, with the remaining group reserved for the test set. The test group is then rotated through.

For testing, we use the same pruning methodology as above, but where necessary, further prune the candidate utterance-proposition pairs from the test set using a top-k pruning strategy, for which we use the previously trained cross-encoder. Specifically, we compute the cosine similarities between the embeddings of an utterance and the remaining candidate propositions, while interchanging their mutual positions. For instance, if (u_i, p_j) represents an utterance-proposition pair, we encode both $[V_{u_i}, V_{p_j}]$ and $[V_{p_j}, V_{u_i}]$ to retain their positional information. Since the cross-encoder has been trained to minimize the mean of the bidirectional BCE loss, the latent representations of positive pairs likely point in similar directions in the embedding space vis-à-vis the negative pairs. As such, a top-k pruning strategy allows us to generate the most similar candidate propositions for a particular utterance *and remove more obvious mismatches.* This helps the system's precision by minimizing the loss of pairs during pruning. We use $k = 5$ to ensure approximate consistency with the training set, which has a 1:4 ratio of positive to negative samples. We then score these leftover pairs using our trained cross-encoder. For each utterance, we consider the extracted proposition to be the one with the highest score as given by the cross-encoder since need a ranking system to choose a proposition for the evaluation metrics.

Cosine similarity. For a given utterance's vector representation, we compute the cosine similarities between the embeddings of all candidate propositions and the utterance embeddings. We then sort these cosine similarities, retrieving the proposition(s) with the most similar embeddings to the utterance embedding. While some level of pruning is required to keep training the cross-encoder tractable due to the quadratic complexity of training a pairwise scorer, for the cosine similarity method we also evaluate against an unpruned candidate set as no training of a separate model is required and the cosine similarity scores can be cached with a single pass. Because cosine similarity calculations only require the utterances to be encoded through a pre-trained model, and no training of a separate model, we simply compare the encodings of utterances to those of propositions without the need for a leave-one-group-out split.

5. RESULTS

We report intersection over union (IOU) scores, top-1, and top-3 accuracy. All 3 metrics used are standard metrics for evaluating retrieval systems (e.g., in computer vision and NLP). The IOU metric allows "partial credit." Since our task is to extract the proposition from a transcript, we calculate the overlap between the extracted proposition and the true proposition. For example, if the true proposition is $red = 10 \wedge blue = 20 \wedge green = 10$ and we extracted proposition $red = 10 \wedge blue = 30$, we consider the cardinality of the intersection of the two sets ($\{red = 10\}$) over their union ($\{red = 10, blue = 20, green = 10, blue = 30\}$). This assesses partial matches where some, but not all, of the correct propositional content is retrieved. Accuracy is a more restrictive metric because it requires exact matches. Top-3 accuracy requires exact matches but is considered correct if the match falls in the top 3 retrievals.

We report results on both the cross-encoder (averaged across all test groups) and the cosine similarity method, at the three different levels of data cleaning discussed in Section. 4.1. For the cosine similarity method, we report results with and without heuristic pruning of the candidate propositions. All results are given in Tables 1–3.

The different segmentation and transcription methods and

Table 1: Cross-encoder performance averaged across test groups. LF represents the pretrained Longformer model.

	Level 1 Cleaning						Level 2 Cleaning						Level 3 Cleaning					
	Oracle ($n = 115$)			Google ($n = 110$)			Oracle ($n = 89$)			Google ($n = 76$)			Oracle ($n = 76$)			Google ($n = 61$)		
	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF
IOU	.526	.448	.457	.353	.383	.281	.596	.585	.525	.537	.530	.374	.664	.683	.608	.635	.645	.399
Acc.	.496	.426	.426	.309	.336	.255	.562	.573	.494	.526	.500	.355	.640	.671	.573	.607	.607	.377
Top-3	.609	.557	.557	.427	.464	.391	.730	.798	.742	.737	.697	.597	.773	.829	.773	.787	.738	.607

Table 2: Cosine similarity performance with heuristic pruning. LF represents the pretrained Longformer model.

	Level 1 Cleaning						Level 2 Cleaning						Level 3 Cleaning					
	Oracle ($n = 115$)			Google ($n = 110$)			Oracle ($n = 89$)			Google ($n = 76$)			Oracle ($n = 76$)			Google ($n = 61$)		
	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF
IOU	.419	.229	.296	.164	.036	.167	.505	.284	.379	.232	.052	.230	.570	.337	.415	.281	.057	.270
Acc.	.374	.200	.278	.144	.027	.162	.472	.258	.359	.210	.039	.223	.547	.307	.400	.262	.049	.262
Top-3	.514	.356	.417	.198	.081	.252	.651	.461	.528	.276	.118	.355	.747	.520	.587	.344	.147	.409

Table 3: Cosine similarity performance without heuristic pruning.

	Level 1 Cleaning						Level 2 Cleaning						Level 3 Cleaning					
	Oracle ($n = 115$)			Google ($n = 110$)			Oracle ($n = 89$)			Google ($n = 76$)			Oracle ($n = 76$)			Google ($n = 61$)		
	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF	BERT	RoBERTa	LF
IOU	.137	.189	.067	.042	.031	.036	.169	.232	.076	.062	.046	.046	.200	.269	.063	.069	.049	.042
Acc.	.113	.148	.060	.036	.027	.027	.146	.191	.067	.053	.039	.039	.173	.227	.053	.065	.049	.033
Top-3	.191	.217	.122	.072	.045	.063	.247	.281	.124	.105	.066	.092	.293	.307	.107	.131	.082	.082

different levels of data cleaning result in different numbers of utterances across the entire dataset in different experimental conditions. These are given as the values of n in Tables 1–3.

Comparison of data cleaning strategies. As expected, with increased levels of data cleaning, we see a trend of improving performance across all extraction strategies, language models, and transcription methods. A larger increase in performance is observed when comparing cosine similarity with pruning to cosine similarity without it, as the pruning strategy especially targets the high proportion of negative matches for a given utterance. Increased cleaning also comes at the cost of fewer samples to evaluate on.

Comparison of extraction methods. The cross-encoder consistently outperforms the cosine similarity baseline across all three metrics. Comparing Table 1 to Table 2 shows that the cross-encoder outperforms the cosine baseline (with heuristic pruning) by at least .2 IOU on average. On the other hand, with a metric that does not reward partial selection, like traditional accuracy, the cross-encoder outperforms the cosine baseline by at least 40%, on average, although the absolute scores are typically lower than the more lenient IOU metric.

Comparison of transcription methods. As expected, using automatic transcriptions of the speech leads to a consistent degradation in performance, as automated segmentation and transcription may incorrectly conflate two overlapping utterances from different people, or leave out or insert words, where such errors are expected to be minimized by careful human annotators. However, this degradation can sometimes be quite small, especially at higher levels of data cleaning, when using the cross-encoder, and the BERT or RoBERTa models. For instance, when using the cross-encoder, the accuracy using BERT embeddings of Google transcriptions increases from 30.9% at data cleaning Level 1 (least stringent) to 60.7% at Level 3 (most stringent), while when using cosine similarity with pruning, accuracy only increases from 14.4% to 26.2%.

Comparison of language models. Using embeddings from BERT typically achieves the best performance, but the performance gap with RoBERTa embeddings is usually quite small especially for the cross-encoder. Both of these models significantly outperform the Longformer model. This may seem surprising at first because the Longformer model is a standard in the coreference approaches we adapted the cross-encoder from, but the Longformer model is optimized to handle large inputs such as entire documents, and in fact appears to underperform on the short utterance transcriptions we use here. For instance, the average length of Oracle utterances in the dataset after Level 1 data cleaning is 45.12 words ($\sigma = 33.34$), and the shorter context window of BERT or RoBERTa may be better equipped to handle these than Longformer, with a context length of 4,096 tokens.

Across all levels of data cleaning, the performance difference between the cross-encoder and cosine similarity is minimized when using BERT. For instance, the cosine similarity with the pruning method's IOU score is only .1 behind the cross-encoder when both use the BERT encoder. On the other hand, the difference in IOU performance between the two methods when using the Longformer model is around $\approx$.14 IOU when averaged across all levels of cleaning and across the Oracle and Google transcripts. Since BERT's `[CLS]` token captures the outcome of the next sentence prediction pretraining objective (which RoBERTa or Longformer do not use), and since the cosine similarity method uses this token as the provenance token for classification, it is possible that cosine similarity with this BERT token better captures the innate sequential coherence in the utterances than the other base models.

Moreover, the parameters of pre trained LMs like BERT are learned in a self-supervised way to reflect human-created data such as Wikipedia [11]. Therefore, the cosine similarity baseline, which undergoes no further fine-tuning and thereby retains its original distribution, performs comparatively better on human-labeled utterances compared to the automated Google transcripts. On the other hand, the lower performance drop for the cross-encoder, especially for stricter pruning strategies compared to more lenient ones suggests that this method, though robust to various levels of candidate sampling, still necessitates a trade-off between sampling cost and performance, especially in real-life applications where automated transcripts are more likely to be seen at inference.

In general, the above trends suggest that supervised training with cross-attentional signals is crucial for consistent performance in the proposition extraction task, as revealed by the performance difference across various levels of cleaning, models, and whether utterances are transcribed by humans or automatically. While the cosine baseline is a relatively low-cost procedure since we only need to run one pass to compute the pairwise similarities, the cross-encoder framework is more generalizable and better performing across domain shifts (Google vs. Oracle utterances), especially for tasks like proposition retrieval from unstructured interactions. More importantly, compared to the cosine-similarity method that requires a full n^2 squared pairwise computations at inference, the cross-encoder operates at a linear complexity $(n * k)$ as long as $k \ll n$, since the same encoder sequentially prunes all but the top-k highest scoring candidates to make its decisions. This is important for such retrieval systems at scale.

6. DISCUSSION

Figs. 4 and 5 show IOU and top-3 accuracy results from the test samples of each group, at Level 1 (most lenient) data cleaning, using BERT embeddings. The plots compare performance using Oracle vs. Google utterances and compare the cross-encoder to cosine similarity with heuristic candidate pruning.

We can see that cross-encoder performance on Group 7 is nearly identical regardless of which transcription method was used. This is likely because Group 7's utterances used mainly simple propositions of the form <color> <relation> <weight>. These instances are easy to extract from the transcripts, and the automated transcripts are likely of high-fidelity.

We can see in Fig. 4 that Group 4's IOU drops significantly when comparing cosine similarity's performance over Oracle transcriptions vs. over Google transcriptions. While exploring the samples from this group, several issues were noted. We found 8 utterances in the Oracle data and only 7 in the Google data, meaning that one of the utterances was completely missed by Google ASR. This utterance happened to be very straightforward and easy for the cosine method to classify. The Oracle transcript is simply "blue ten." Another issue, again due to the segmentation, is Google ASR may merge 2 utterances. This highlights a limitation of ASR models, where some additional context may needed to know when a speaker has moved to another sentence. Obviously, the main difference between using the different transcription methods is the transcripts themselves. One instance from Group 4 states "easy green block twenty cause ..." whereas Google ASR transcribed the utterance as "okay e green block red block 10 ...". These results highlight certain issues that should be considered when deploying such an information extraction system over the outputs of an ASR system, as may be required in classroom environments.

6.1 Error Analysis

As the cross-encoder is consistently the best-performing extraction method, examining samples it gets wrong is informative. One such example is the utterance "green block one probably twenty ten ten twenty". The correct proposition is $blue = 10 \land green = 20 \land red = 10$. The annotators have access to the video and can see that when saying "ten ten twenty," the speaker is actually pointing to the blue block, then the red block, then the green block. This information is not available through the textual medium alone.

Top-k Errors. In order to compare our two extraction methods, we carried out a detailed analysis of candidate propositions that were ranked similarly, based on their cross-encoder scores or cosine similarities. On average, at Level 1 (most lenient) data cleaning, the cross-encoder performs comparatively better at ranking the correct propositions in the top 5. For instance, the cross-encoder ranks 8 and 21 correct propositions higher than the cosine similarity method, for Google and Oracle transcripts respectively. The cosine similarity method ranks 1 (Google) and 11 (Oracle) correct propositions higher. On the other hand, there were at least 14 Google utterance transcripts and 37 Oracle utterance transcripts where both the extraction methods performed equivalently.

Qualitative Analysis. On average, simpler utterances that contain a reference to only one color and/or weight are correctly retrieved by both the cross-encoder and cosine similarity. For instance, "I tell red cube ten grams" (correct proposition $red = 10$) and "green twenty" ($green = 20$). More interestingly, the cross-encoder seems to retrieve utterances with ambiguous context without a direct reference to color or with multiple colors more effectively than the cosine similarity method. For example, "Fifty I" ($yellow = 50$) and "green block twenty red block, blue block ten ten" ($blue = 10 \land green = 20 \land red = 10$). This is likely due to the cross-encoder's cross-attention based signals that are being sourced from the entire utterance in the context of the candidate proposition. This was previously observed in [5] where modeling global signals in parallel with local features led to an overall increase in coreference resolution performance.

7. LIMITATIONS

With the data cleaning procedures comes an inevitable loss of several utterances. The result is a small dataset, ranging

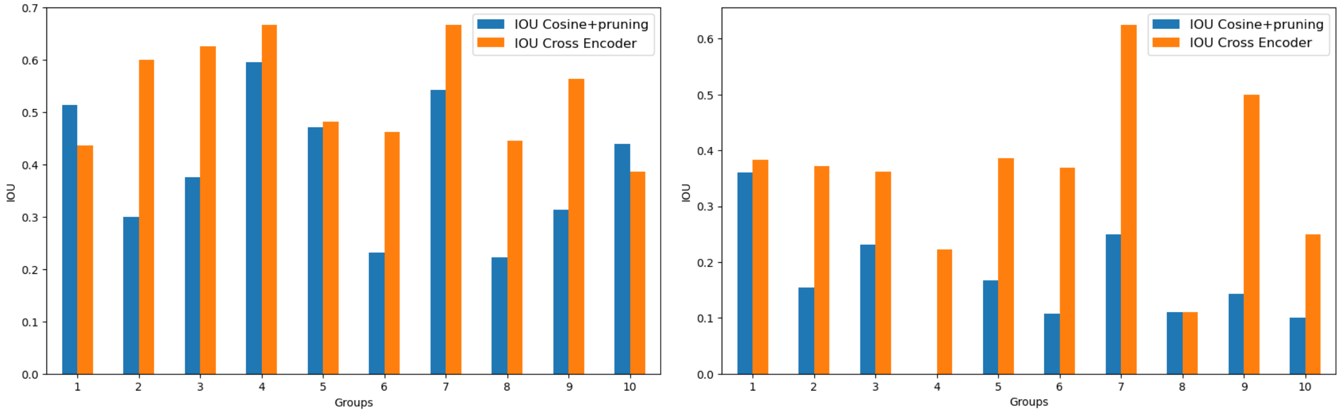

Figure 4: Group-wise IOU at Level 1 (the least restrictive) data cleaning using BERT. Performance with Oracle transcriptions is given on the left and performance with Google transcriptions is given on the right.

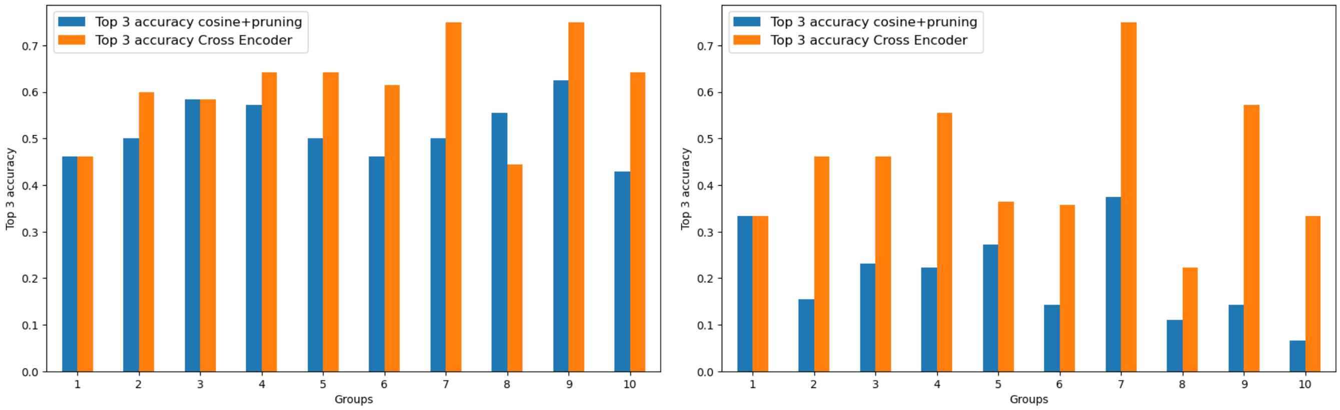

Figure 5: Group-wise top-3 accuracy at the Level 1 (the least restrictive) data cleaning using BERT. Performance with Oracle transcriptions is given on the left and performance with Google transcriptions is given on the right.

from dozens to slightly over 100 utterances, depending on the level of data cleaning.

Errors in the automated transcripts can adversely affect the efficacy of the candidate pruning process, since there are errors in the way it transcribes the colors and weights. For example, Google transcribes an utterance as “blue block ’s obviously time”, when clearly “time” should have been transcribed as “10”. This has an impact on the pruning for candidate propositions since pruning of candidates relative to this transcription will look for all propositions that mention “blue” instead of “blue” *and* “10”.

The heuristic pruning of candidate propositions has a large effect on the performance as seen in the performance of cosine similarity with and without pruning. Pruning significantly reduces the search space and can be partially credited with a lot of performance improvement, including that of the cross-encoder, since pruning is baked into the method to maintain a more balanced sample distribution for training and to keep the test data resembling the distribution of the training data. However, the pruning methodologies must be adapted to the nature of the propositions in the task and so is not automatically generalizable.

Finally, participants in the Weights Task Dataset consented to recording and analysis via third-party tools, such as Google ASR, for research purposes. To protect student privacy, a real classroom implementation would need to use a local or custom model to avoid the ethical implications of sending student data to a private company’s servers.

8. CONCLUSIONS

In this paper, we have defined and explored the complex problem of automatically identifying propositional content from transcriptions of natural speech in a collaborative task. Automated propositional extraction from speech serves a number of important educational purposes. For example, tracking the assertion of propositions over time indicates how students are/are not discussing key concepts relevant to the task, which in turn indicates the construction of shared knowledge [35].

The Weights Task data presents many challenges, from overlapping speech to incomplete sentences, and we have evaluated a suite of transformer-based language models based on two different methodological frameworks: a cosine similarity baseline vs. a cross-encoder. Our experiments present a feasible method for performing the extraction of task-relevant propositions by building upon publicly-available language models and pairwise representation learning techniques. Additionally, our best performing methods particularly the cross-encoding framework show a narrow performance gap when operating over automated transcriptions when compared to human "Oracle" transcriptions, suggesting a feasible path forward toward fully automating such a system in a live environment. A clear application in a classroom is in a system that models the shared knowledge of a group toward the task goal, and might be a component of an AI agent who assists small groups in collaborative problem solving (CPS) [13].

In order to generalize to other domains, we need only an inventory of task-relevant propositions, which can be enumerated deterministically as in Sec. 4. Ground-truth annotation is needed for cross-encoder training, but our success on a small amount of data demonstrates the small amount of needed annotation.

9. FUTURE WORK

Our work in this paper has been conducted over transcription of speech only, however in a multimodal dataset, multimodal features play a significant role in interpreting the dialogue and discourse. Therefore the addition of multimodal features such as gestures, actions, or detected objects in video have the potential to significantly improve performance.

The dense paraphrasing procedure (Section 3.1) is one way of enriching the textual channel with information from other channels. This is partially automated already, but full automation would represent another step toward a live deployable system. This would involve focusing on multimodal anaphora decontextualization, where the goal is to disambiguate pronouns by associating them with specific referents in video segments, involves identifying pronouns and their antecedents, and linking these antecedents to visual elements in a video. This would require minimally the following steps: Coreference resolution on pronouns and entity referents, which could use a similar cross-encoder architecture as we use herein; pronoun identification and semantic analysis; paraphrasing pronouns with entity referents based on the context provided by overlapping objects and actions in the utterances; video analysis for referent identification, such as object detection and recognition.

Some technical improvements to things like segmentation, such as through advanced speaker diarization [27] would also alleviate some of the difficulties caused by automated transcripts. Customizations to the cross-encoder training, such as through the inclusion of a contrastive loss or with a set of challenging negative pairs (*a la* [6] or [16]) could assist with smarter pruning strategies. The addition of global features, such as by modeling utterances as the proposition level could also be a step toward an end-to-end model—potentially one that does not require the use of predefined heuristics.

Finally, applying our methods to another dataset, such as the Wason DeliData [22], will provide further insights into the robustness of our methods decoupled from a specific task, and will further demonstrate the feasibility of deploying such a system in real collaborative and classroom environments.

10. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation (NSF) under subcontracts to Colorado State University and Brandeis University on award DRL 2019805 (Institute for Student-AI Teaming), and by Other Transaction award HR00112490377 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program. Approved for public release, distribution unlimited. Views expressed herein do not reflect the policy or position of the National Science Foundation, the Department of Defense, or the U.S. Government. All errors are the responsibility of the authors. Our thanks to the anonymous reviewers whose feedback helped improve the final copy of this paper, and to August Garibay and Carlos Mabrey for extensive data annotation.

11. REFERENCES

[1] S. R. Ahmed, A. Nath, J. H. Martin, and N. Krishnaswamy. $2 * n$ is better than n^2: Decomposing event coreference resolution into two tractable problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1569–1583, Toronto, Canada, July 2023. Association for Computational Linguistics.

[2] S. R. Ahmed, A. Nath, M. Regan, A. Pollins, N. Krishnaswamy, and J. H. Martin. How good is the model in model-in-the-loop event coreference resolution annotation? *arXiv preprint arXiv:2306.05434*, 2023.

[3] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. Automatic extraction of opinion propositions and their holders. In *2004 AAAI spring symposium on exploring attitude and affect in text*, volume 2224, 2004.

[5] A. Caciularu, A. Cohan, I. Beltagy, M. E. Peters, A. Cattan, and I. Dagan. Cdlm: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, 2021.

[6] A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, 2021.

[7] H. Chai and M. Strube. Incorporating centering theory into neural coreference resolution. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2996–3002, 2022.

[8] V. Chand, K. Baynes, L. M. Bonnici, and S. T. Farias. A rubric for extracting idea density from oral language samples. *Current protocols in neuroscience*, 58(1):10–5, 2012.

[9] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[10] S. Dennis. An unsupervised method for the extraction of propositional information from text. *Proceedings of*

the National Academy of Sciences, 101(suppl_1):5206–5213, 2004.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[12] H. Gijlers and T. de Jong. Sharing and confronting propositions in collaborative inquiry learning. *Cognition and Instruction*, 27(3):239–268, 2009.

[13] A. C. Graesser, S. M. Fiore, S. Greiff, J. Andrews-Todd, P. W. Foltz, and F. W. Hesse. Advancing the science of collaborative problem solving. *psychological science in the public interest*, 19(2):59–92, 2018.

[14] B. J. Grosz. Focusing in dialog. In D. L. Waltz, editor, *Theoretical Issues in Natural Language Processing-2*, 1978.

[15] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204, 1986.

[16] W. Held, D. Iter, and D. Jurafsky. Focus on what matters: Applying discourse coherence theory to cross document coreference. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1406–1417, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[17] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[18] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2020.

[19] S. Jeon and M. Strube. Centering-based neural coherence modeling with hierarchical discourse segments. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7458–7472, Online, Nov. 2020. Association for Computational Linguistics.

[20] Y. Jo, J. Visser, C. Reed, and E. Hovy. A cascade model for proposition extraction in argumentation. In B. Stein and H. Wachsmuth, editors, *Proceedings of the 6th Workshop on Argument Mining*, pages 11–24, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[21] Y. Jo, J. Visser, C. Reed, and E. Hovy. Extracting implicitly asserted propositions in argumentation. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 24–38, Online, Nov. 2020. Association for Computational Linguistics.

[22] G. Karadzhov, T. Stafford, and A. Vlachos. Delidata: A dataset for deliberation in multi-party problem solving. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW2):1–25, 2023.

[23] I. Khebour, R. Brutti, I. Dey, R. Dickler, K. Sikes, K. Lai, M. Bradford, B. Cates, P. Hansen, C. Jung, B. Wisniewski, C. Terpstra, L. Hirshfield, S. Puntambekar, N. Blanchard, J. Pustejovsky, and N. Krishnaswamy. When text and speech are not enough: A multimodal dataset of collaboration in a situated task. *Journal of Open Humanities Data*, 2024.

[24] I. Khebour, K. Lai, M. Bradford, Y. Zhu, R. Brutti, C. Tam, J. Tu, B. Ibarra, N. Blanchard, N. Krishnaswamy, et al. Common ground tracking in multimodal dialogue. In *Proceedings of the 1st Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, 2024.

[25] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[26] J. Li, M.-T. Luong, and D. Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115, 2015.

[27] Z. Li and J. Whitehill. Compositional embedding models for speaker identification and diarization with simultaneous speech from 2+ speakers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7163–7167. IEEE, 2021.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[29] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In S. Riezler and Y. Goldberg, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[30] A. Nath, S. Manafi, A. Chelle, and N. Krishnaswamy. Okay, let's do this! modeling event coreference with generated rationales and knowledge distillation. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 2024.

[31] A. Nath, S. Mannan, and N. Krishnaswamy. AxomiyaBERTa: A phonologically-aware transformer model for Assamese. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11629–11646, Toronto, Canada, July 2023. Association for Computational Linguistics.

[32] E. Pacuit. *Neighborhood semantics for modal logic.* Springer, 2017.

[33] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[34] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[35] J. Roschelle and S. D. Teasley. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer, 1995.

[36] C. Sun, V. J. Shute, A. Stewart, J. Yonehiro, N. Duran, and S. D'Mello. Towards a generalized competency model of collaborative problem solving. *Computers & Education*, 143:103672, 2020.

[37] J. Tu, K. Rim, E. Holderness, B. Ye, and J. Pustejovsky. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS)*, Nancy, France, June 2023. Association for Computational Linguistics.

[38] J. Tu, K. Rim, and J. Pustejovsky. Competence-based question generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1521–1533, Gyeongju, Republic of Korea, Oct. 2022. International Committee on Computational Linguistics.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[40] N. M. Webb, M. Ing, E. Burnheimer, N. C. Johnson, M. L. Franke, and J. Zimmerman. Is there a right way? productive patterns of interaction during collaborative problem solving. *Education Sciences*, 11(5):214, 2021.

[41] X. Yu, W. Yin, and D. Roth. Pairwise representation learning for event coreference. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 69–78, 2022.

[42] Y. Zeng, X. Jin, S. Guan, J. Guo, and X. Cheng. Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3084–3094, 2020.

[43] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert, 2020.