Choose Your Words Wisely: Leveraging Embedded Dialog Trajectories to Enhance Performance in Open-Domain Conversations

Nancy Fulda(✉), Tyler Etchart, and Will Myers

Brigham Young University, Provo, UT 84602, USA
{nfulda,tyler.etchart,william_myers}@byu.edu
http://DRAGN.ai

Abstract. Human conversations are notoriously nondeterministic, and identical conversation histories can nevertheless accept dozens, if not hundreds, of distinct valid responses. In this paper, we present and expand upon *Conversational Scaffolding*, a response scoring method that capitalizes on this fundamental linguistic property. We envision a conversation as a set of trajectories through embedding space. Our method leverages the analogical structure encoded within language model representations to prioritize possible conversational responses with respect to these trajectories. Specifically, we locate candidate responses based on their linear offsets relative to the scaffold sentence pair with the greatest cosine similarity to the current conversation history. In an open-domain dialog setting, we are able to show that our method outperforms both an Approximate Nearest-Neighbor approach and a naive nearest neighbor baseline. We demonstrate our method's performance on a retrieval-based dialog task using a retrieval dataset containing 19,665 randomly-selected sentences. We further introduce a comparative analysis of algorithm performance as a function of contextual alignment strategy, with accompanying discussion.

Keywords: Response prioritization · Utterance retrieval · Word embeddings · Conversational AI

1 Overview

The one-to-many hypothesis of dialog as explored by Zhao et al. [30] asserts that the correct next response to for a given dialog history is not dependent on the dialog history alone, but is instead a function of many variables related to user state, world state, and dialog state, and that it is nontrivial to extract them all. This leaves connectionist approaches to dialog modeling with an interesting predicament. How can one apply a classic training paradigm to dialog modeling when there is no single right answer to the question of "what should I say next?"

A. P. Rocha et al. (Eds.): ICAART 2020, LNAI 12613, pp. 236–253, 2021.
https://doi.org/10.1007/978-3-030-71158-0_11

A commonly-applied solution includes the use of variational autoencoders [2,21,26,28,30], which model the unknown conversational elements as a stochastic process. While often effective, this method is data-hungry, and high-quality conversational data is scarce. Consequently, it is often necessary to train the language models on larger datasets of lesser quality, such as lightly pre-processed exchanges from online chat forums, rather than on small but high quality datasets.

In [27] we presented an alternate approach: a *Conversational Scaffolding* method that leveraged the conversational patterns in a small, high-quality scaffold corpus in order to rank candidate responses in a retrieval-based conversation system. This paper expands and improves upon that work by presenting a more detailed analysis of the scaffolding approach as well as a comparative study of algorithm performance as a function of contextual alignment method. We frame the conversational scaffolding method in terms of dialog trajectories, with linear offsets between the embedded representations of context sentences used in order to find the most appropriate match.

A key advantage of this approach is its ability to leverage the power of connectionist systems while still adhering to the conversational norms and patterns exemplified by a small, highly curated dataset. Specifically, the language model used to embed context sentences is a classic connectionist system trained on large scale, broad-topic text corpora, and is able to leverage the inherent linguistic knowledge common to such models during the embedding process. Once each sentence has been localized within an 512-dimensional linguistic space, however, the process of analyzing the dialog history and scoring candidate responses is handled via a low-resource algorithm.

1.1 The Challenge of Large Conversational Datasets

While the internet era has provided unprecedented access to large-scale text corpora across a variety of styles and topics, high-quality conversational data is more difficult to come by. Unmoderated online interactions, while plentiful and easy to harvest, often fail to exhibit the topical continuity, common courtesy, and social restraint that one might like to replicate in an automated chat system. This is caused not only by the unfortunate prevalence of trolling [6,9,24], but also by the varying personalities, ideologies, and social competencies of the conversation participants themselves. Dialogs extracted from movie scripts [32] or technical support forums [18] are often more coherent, but fail to exhibit the conversational patterns of common, everyday speech. (It would seem odd indeed if an automated personal assistant trained using such data were to provide unsolicited technical advice, or profess its newly-discovered love toward its conversation partner).

A further complication arises from the crowd-sourced nature of all such online conversational data. The life histories, demographics, political opinions, and general likes and dislikes of the conglomerate chatters are so disparate that any language model trained using them is almost guaranteed to seem schizophrenic, producing mutually incompatible statements with distressing frequency.

We seek to alleviate these challenges by introducing a conversational response-scoring method that does not require such massive amounts of data, and can instead be used in conjunction with a relatively small scaffold corpus. Our method also does not require any network updates or fine-tuning of the Incorporated language models, and is often able to respond appropriately to conversation histories that are not well represented in the scaffold corpus because it relies, not on the specific embedding locations of individual sentences, but rather on their locations relative to one another. It is the general *patterns* of language, the dialog trajectories that describe the transition from question to answer and back again, that we seek to emulate.

1.2 The Analogical Structure of Embedding Spaces

In addition to generating and categorizing text, it has become increasingly popular to extract the hidden layer activations of large language models for the purpose of semantic evaluations. Favorite models for this purpose include Sentence-level embedding spaces such as skip-thought vectors [15], quick-thought vectors [17], InferSent [10], and Google's Universal Sentence Encoder [8], as well as contextualized word embedding models such as BERT [12] and ELMo [23].

A driving force behind this tendency is the phenomenal and fascinating ability of word-level embedding spaces to encode human-interpretable knowledge in the relative locations of embedded texts. For example, the word2vec [19], GLoVE [22], and FastText [4] models can be used to solve linguistic analogies of the form *a*:*b*::*c*:*d*. This is generally accomplished using vector offsets such as [*Madrid - Spain + France* ≈ *Paris*] or [*walking - walked + swimming* ≈ *swam*] [14,20]. The sums and differences between the embedded representations of the first three words are calculated, and then a nearest-neighbor search across the model's vocabulary (excluding the three source words) produces the solution to the analogical query.

Our research extends this notion of analogical relationships into the realm of multi-word embeddings. We postulate (and show via our results) that sentence-level embedding spaces can contain similar analogical relationships, and that these relationships can be utilized to select plausible responses in open-domain dialogs. Thus, rather than evaluating candidate responses based on their strict distance to exemplars in the scaffold corpus, we instead rely on the relative distance between pairs of sentences in order to locate an idealized response vector which corresponds to point *d* in the classic *a*:*b*::*c*:*d* analogical form. Candidate responses are scored based on their cosine distance from this target point.

2 Embeddings, Scaffolds, and Dialog Trajectories

We begin our response-ranking process by encoding a reference corpus, called our *scaffold*, using one of many available pre-trained embedding models. Incoming utterances are matched against the scaffold corpus, and the top n contextual matches are used to calculate an analogically coherent response, or *target point*

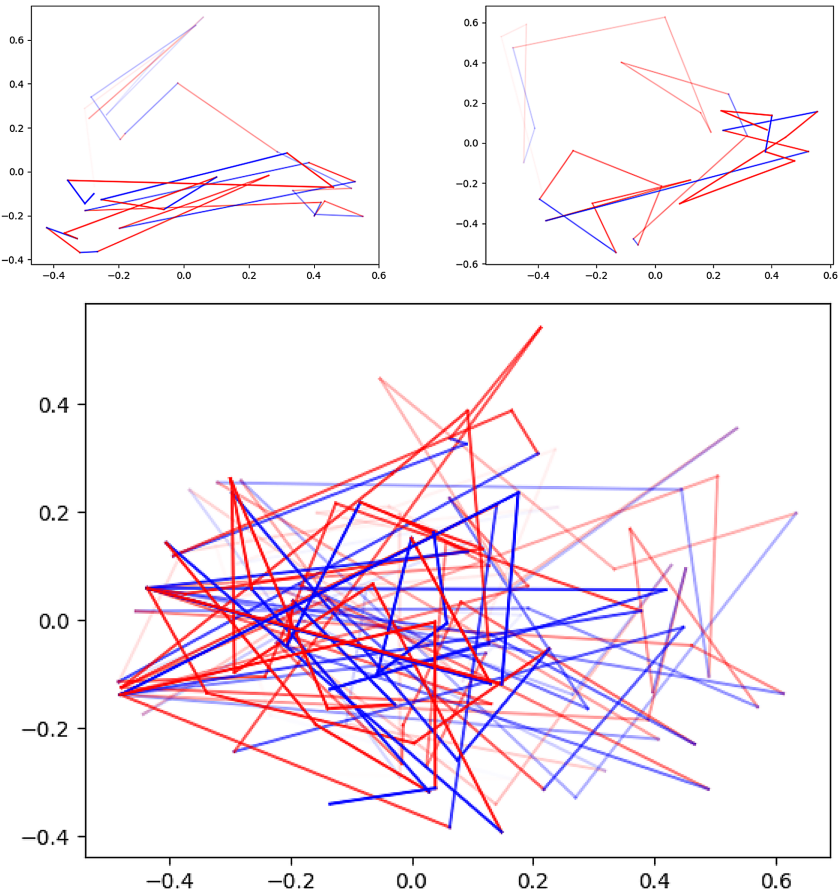

Fig. 1. PCA Reduction: Dialog trajectories from three conversations in the Chit-Chat Dataset, encoded using Google's universal sentence encoder lite [8]. Blue lines indicate chat partner 1, red lines are chat partner 2. Alpha values correspond to the index of each utterance over time, with faint lines representing earlier messages and dark lines representing later ones. The objective of our scaffolding algorithm is to find subpaths within the scaffold corpus which match the general location and trajectory of the current dialog history, then user the next point on the highest-ranking subpaths to select the best response candidate. (Color figure online)

within the embedding space. The candidate response with the lowest cosine distance from the target point is selected as our agent's dialog output.

Figure 1 shows sample dialog trajectories from three conversations. In each, it is possible to observe the meandering of the conversation topic and utterance type over time. Critically, one can observe a certain tendency toward repetition, both within each plot and between the plots as a whole, and this behavior is equally visible when dimensionality reductions other than PCA are used. In short, while language is combinatorial in nature and thus able to represent a

nearly infinite span of ideas, the *patterns* of language are far more tractable. Certain types of statements encourage certain types of responses, regardless of the specific conversation topic. These patterns can be detected and imitated via the use of analogical relations within a pre-trained embedding space. Thus, a relatively small corpus of exemplars can be used to guide the response ranking system of a conversational agent.

After due consideration, we selected Google's Universal Sentence Encoder Lite [8] as the embedding model of choice for this application. This decision was based primarily on its unusually high performance as a heuristic for semantic distance. Experiments in our laboratory revealed that USE Lite was able to achieve a Pearson's r score of 0.751 on the 2017 Semantic Textual Similarity benchmark, the highest score of any model we tried, as shown in Table 1. It is possible that Google's large model would have performed even better, but exploratory applications found the large model too slow to implement on a sentence-by-sentence basis. In a real-time conversational scenario, there is no possibility of batch-processing utterances, and so we opted to consider only those models which might reasonably be employed in a real-world setting.

Table 1. Model performance on the SemEval 2017 Semantic Textual Similarity Benchmark [7] and the Stanford Natural Language Inference Corpus [5] evaluated using Pearson's r and Spearman's rho (higher is better). The greatest value in each column is shown in bold-face text.

	STS r	STS rho	SNLI r	SNLI rho
GPT-2	−0.052	0.092	−0.007	0.019
InferSent	0.718	0.702	0.273	0.279
Google use lite	**0.751**	**0.737**	**0.366**	**0.367**
Transformer-XL	0.341	0.341	0.112	0.112
Skip-thought	0.214	0.296	0.046	0.108
BERT BoW	0.495	0.490	0.166	0.174
FastText BoW	0.547	0.543	0.248	0.257
Glove BoW	0.404	0.440	0.241	0.247

2.1 Conversational Scaffolding

Conversational Scaffolding [27] is a response-ranking algorithm that relies on the structural properties of an analogically coherent embedding space in order to select high-quality candidates from a repository of possible responses. In this paper, the embedding space used is that provided by Google's universal sentence encoder lite [8].

Figure 2 gives an overview of our conversational scaffolding algorithm. Given a dialog context of variable length, our algorithm first locates a set of high-quality

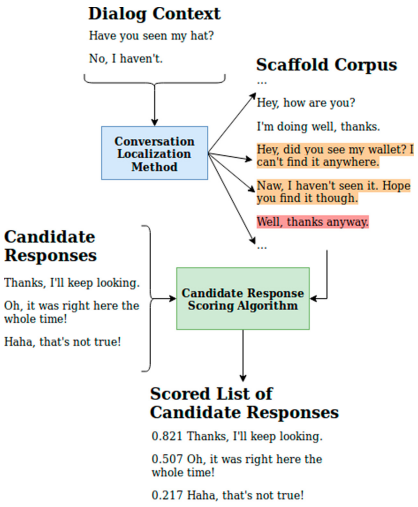

Fig. 2. Workflow diagram: The dialog context is converted to an array of sentence embeddings using Google's Universal Sentence Encoder, then passed to an embedded concatenation localization function to determine the best contextual match(es). The matched utterances (orange) along with their direct successors (red) are then passed to the Response Scoring Algorithm, which assigns a numerical value to each candidate response. Image originally published in [27]. (Color figure online)

contextual matches within the scaffold corpus. These contextual matches, along with the dialog sentence directly following each context match, are then passed to one of several scoring algorithms.

2.2 Contextual Alignment

We use a *contextual alignment* process to match incoming sentences against similar sentence patterns within a scaffold corpus. This can be done naively by using an Approximate Nearest Neighbor algorithm based on a simple Euclidean distance metric. In this paradigm, for a dialog context of length n, the optimal contextual match can be identified as follows:

$$min_z \sum_{i=1}^{n} ||v_i - s_{z+i}|| \tag{1}$$

where $\{v_1, ..., v_n\}$ are the vector embeddings of the n most recent sentences in the current dialog and $\{s_{z+1}, ..., s_{z+n}\}$ represent the vectors located within a sliding window of length n beginning at element z of the pre-embedded scaffold corpus. The notation $||x||$ represents the Euclidean norm of vector x.

This Euclidean distance approach is easy to calculate, but it ignores the powerful analogical structure inherent within the embedding space [27]. In order to capture such subtleties, we compare this approach against two alternate method of contextual alignment: *Embedded Concatenation* and *Difference Vectors*.

Embedded Concatenation. *Embedded Concatenation* leverages the structure of the embedding space by concatenating the input sentences prior to encoding them via Universal Sentence Encoder Lite [8]. A naive Euclidean distance metric is then used to match the embedded concatenation against each element in the pre-embedded scaffold corpus. The optimal contextual match is:

$$min_z \; ||embed(h_1 + ... + h_n) - s_z|| \tag{2}$$

where $\{h_1, ..., h_z\}$ are the plain text (i.e. *un*embedded) utterances in the dialog history, the + symbol represents string concatenation (with an extra space inserted between sentences), s_z is an arbitrary vector located within the pre-embedded scaffold corpus, and $embed(x)$ denotes the process of embedding a plain text utterance x to obtain its corresponding vector representation.

Difference Vectors. The *Difference Vectors* approach embeds each sentence in the dialog history separately, then searches for a contextual match with the smallest average distance across all sentences:

$$min_{\sum_i} \; (\; ||embed(h_{i+1}) - embed(h_i)|| \; - \; ||s_{i+1} - s_i|| \;)^2 \tag{3}$$

Note that the described localization methods assume that only a single, optimal, contextual match is desired. This was done for simplicity. In reality, it is often beneficial to take the k best matches, and in fact many of the scoring algorithms in Sect. 3.2 require $k > 1$. The scattershot diagram in Sect. 2.3 assumes a value of $k = 3$ for clarity. In our empirical experiments, a value of $k = 5$ was used.

2.3 Candidate Response Scoring

In [27] we presented three candidate response scoring algorithms for conversational scaffolding, each of which assumes a set of candidate responses g_i and a recent dialog sentence c. We briefly review these algorithms here.

1. *Naive Analogy.* This algorithm is based on the simplifying assumption that the closest context match within the scaffold dataset must of necessity be paired with an optimal response. (In reality, a scaffold sentence with a slightly larger distance from the user utterance might actually be paired with a superior response; this is addressed in the scattershot and flow vector methods, below). Using a value of $k = 1$, the naive analogy locates the sentence in the scaffold corpus whose embedded representation is closest to the most recent dialog utterance, then follows the conversational trajectory between that sentence and its (embedded) successor in order to find a target point.

2. *Flow Vectors.* The flow vectors approach is based on the idea that conversations tend to "flow" from certain regions of embedding space into others, and that all matching utterance pairs will reflect the same general flow direction. Accordingly, rather than simply taking the most promising dialog, we average

the dialog trajectories from multiple contextual matches and then look for a candidate utterance that lies along the resulting flow direction.

3. *Scattershot.* The scattershot scoring algorithm takes the one-to-many property of language into account by assuming that there are many valid responses for each dialog context, and searches for a candidate response that matches *any* of several high-scoring context matches. In this method, the vector differences between each context match and its respective successor are calculated separately, then added to the vector embedding of the most recent utterance in the dialog history. The result is a set of k target points, each of which represents a possible response. The candidate nearest to *any one* of these targets receives the highest score.

Of these three algorithms, we found in prior work that the *scattershot* algorithm performed most impressively with respect to the baselines as well as to the other conversational scaffolding algorithms. We expand upon that result in this paper by comparing the algorithms across a variety of context lengths and contextual matching strategies, and show that *scattershot* continues to be the strongest method.

3 The Scattershot Algorithm

The *scattershot* algorithm (Fig. 3, Algorithm 1) is based on three key principles: (a) the idea that dialog modeling is a nondeterministic task, and there are many correct responses to a given dialog history, (b) under most dialog control paradigms, the selection of possible response candidates is finite, and (c) in many conversational settings, it is not necessary to find the *optimal* response for a given context; you merely have to be good enough to satisfy the user.

Accordingly, the scattershot method examines a number of trajectories identified in the scaffold conversation as being a good contextual match for the current dialog history, and then extends each trajectory to find the target point, or 'ideal response', that would be implied by this trajectory. The repository of candidate responses is then examined to find the candidate which has the minimum distance of any sentence to any target point, and this response is given the highest ranking. Intuitively, this process can be described as seeking a candidate response that is related to the dialog history in the same way that the scaffold corpus successor is related to the sentences that precede it. It is an extension at the sentence level of the classic A:B::C:D analogy structure commonly used in conjunction with word embeddings [14,20].

3.1 Scattershot Performance as a Function of Contextual Matching

No algorithm exists in isolation. The effectiveness of the scattershot algorithm (and other conversational scaffolding methods) is impacted by the methods used to select the contextual matches to be used as its starting points. We evaluate this impact on a response prioritization task across four datasets, which were cleaned and pre-processed as described in [27].

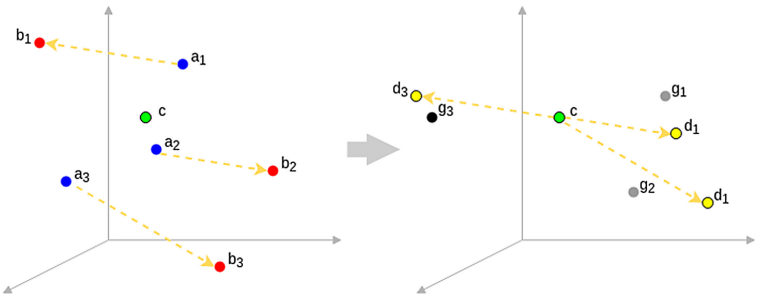

Fig. 3. Diagrammatic depiction of the *scattershot* algorithm for conversational scaffolding. c (green) represents the embedded input sentence, a_i (blue) represent the nearest embedded sentences from the scaffold corpus, b_i (red) represent the associated embedded successors to a_i in the scaffold, $d_i = c + (b_i - a_i)$ (yellow) represent the 'ideal' responses, and g_i (grey and black) represent embedded candidate responses with g_3 (black) representing the response selected by the scattershot scoring algorithm. Image originally from [13]. (Color figure online)

Algorithm 1. Scattershot.

Inputs:
$\boldsymbol{h}$ = Embedded conversation history.
$\boldsymbol{r}$ = Embedded candidate responses produced by the enerators.
$\underset{n \times \|h\|-1}{C}$ = Embedded Chit-Chat dataset

Output:
$S = \{s_1 \ldots s_i \mid s \in [0, 1]\}$ where s_i is the score for r_i

$\boldsymbol{c} \leftarrow [h_1 - h_0, \ldots, h_n - h_{n-1}]$
for i `in 1..5` **do** ▷ Find the n closest points in C to c.
  $\boldsymbol{a_i} \leftarrow min_i(dist(\boldsymbol{u}, C))$ ▷ Where $dist$ is any valid distance metric.
  $\boldsymbol{b_i} \leftarrow$ Find the utterance in C that directly follows $\boldsymbol{a_i}$
  $\boldsymbol{d_i} \leftarrow \boldsymbol{b_i} - \boldsymbol{a_i} + \boldsymbol{c}$ ▷ Where d_i is the "ideal" response vector to $\boldsymbol{b_i}$.
end for
for r_i `in` $\boldsymbol{r}$ **do**
  $\boldsymbol{g_i} \leftarrow min(dist(\boldsymbol{r_i}, \boldsymbol{d}))$
end for
return $1.0 - \frac{g}{\|g\|}$

The response prioritization tasks was set up as follows. 13,244 windowed conversations were selected from four text corpora, with equal representation from each corpus:

1. Chit-Chat[1] [27]
2. Daily Dialog[2] [16]
3. A 33 million word subset of Reddit[3] [25]
4. Ubuntu Dialogue Corpus[4] [18]

[1] https://github.com/BYU-PCCL/chitchat-dataset.
[2] https://aclanthology.coli.uni-saarland.de/papers/I17-1099/i17-1099.
[3] http://files.pushshift.io/reddit/.
[4] https://www.kaggle.com/rtatman/ubuntu-dialogue-corpus.

The Chit-Chat dataset, collected locally via a university competition, contains 483,112 dialog turns between university students using an informal online chat framework. The Daily Dialog dataset simulates common, real-life interactions such as shopping or ordering food at a restaurant. Reddit[5] covers an array of general topics, with copious instances of web links, internet acronyms, and active debate. Finally, the Ubuntu Dialogue Corpus contains 966,400 dialog turns taken from the Ubuntu Chat Logs, with a heavy emphasis on troubleshooting and technical support.

We then set aside 3,311 conversations (about 5% of the smallest corpus) from each dataset to create the evaluation corpus, with the unused portions of each dataset remaining as scaffolding. The scaffold corpus was embedded using Google's Universal Sentence Encoder lite [8], an embedding algorithm that strikes a strong balance between semantic coherence and speed of computation.

We then windowed the evaluation corpus so that rather than 3,311 long conversations, we instead had 13,244 windowed conversations with 5 dialog turns each. Each dialog from this windowed evaluation set was paired with six candidate responses: (a) the correct follow-on sentence for the given dialog context, and (b) five distractors randomly chosen from the same text corpus as the correct answer. The scaffolding algorithms in Sect. 2.3, along with several baselines described in Sect. 3.1, were tasked with identifying the true response.

Baselines. We selected three baselines to compare along with our conversational scaffolding algorithms, the objective being to determine whether performance improves when conversational trajectories are taken into consideration.

Naive Nearest
This algorithm naively selects the successor of the best context match as the 'ideal response' or target point. In other words, rather than calculating the ideal response as $d_1 = c + b_i - a_i$, the naive-nearest algorithm calculates $d_1 = b_i$. This approach ignores the analogical nature of language by assuming that the successor to the best context match represents an optimal response, even if the contexts do not match exactly.

Approximate Nearest Neighbor (ANN)
This algorithm implements an Approximate Nearest Neighbor scoring strategy. Its ideal target point is calculated in the same way as the flow vectors algorithm, but with $d_1 = 1/n \sum b_i$. The impact of conversational trajectories are ignored, and the algorithm instead orients itself based on the successor utterances extracted from the scaffold corpus.

Random
This baseline randomly selects one of the candidate responses without reference to the dialog history. As there are six candidate responses for each evaluation

[5] Due to the massive size of Reddit, we only used a subset of the comments and posts from June 2014 to November 2014.

dialog, only one of which is correct, we can expect the random algorithm to perform with an accuracy of approximately $\frac{1}{6} \approx 16.7\%$.

Neural Network
We also implemented a multilayer regression network using Tensorflow [1]. As input it accepts two utterances from the dialog history, each embedded as a 512 dimensional vector using the Universal Sentence Encoder Lite. It then predicts the ideal target point as a 512 dimensional output vector. The hidden layer sizes were 2048 and 2014 units respectively, with exponential linear unit activation functions, MSE loss, and 25% dropout.

Table 2. Retrieval accuracy on a dialog control task with 13,244 distinct conversations. A context size of $n = 2$ was used for the dialog history. The highest-scoring algorithm in each column is shown in bold-face text. The neural network baseline was unable to select a response for embedded concatenation because it requires two distinct vectors as input, and embedded concatenation provides only one reference vector.

	euclidean dist.	diff. vectors	embed. concat.	average
scattershot	59.30%	**63.99%**	**68.07%**	**63.79%**
flow vectors	60.41%	61.53%	62.47%	61.47%
naive-analogy	56.16%	61.88%	62.29%	60.11%
naive-nearest	56.15%	36.45%	58.97%	50.52%
ANN classifier	**65.54%**	37.71%	64.96%	56.07%
network	50.41%	50.41%	n/a	33.61%
random	17.13%	16.60%	16.06%	16.60%

Results. Experimental results are shown in Table 2. We note with interest that the contextual matching method chosen has a high impact on all scoring algorithms except for flow vectors and the neural network baseline. With a peak accuracy of 68.07% when paired with embedded concatenation, the scattershot algorithm shows a clear advantage over all other methods, outperforming the nearest baseline by 2.53%. We believe that this is because scattershot takes the one-to-many nature of language into account, allowing the system to select a candidate that closely matches one of many possible valid responses.

It is also useful to compare our naive-analogy algorithm with the naive-nearest baseline. These two algorithms are identical except for their analogical content. Our results show that leveraging the inherent analogical properties of the embedding space results in an overall accuracy improvement of 3.32% when using embedded concatenation and by 25.43% when using difference vectors. Additionally, we observe that the naive-analogy algorithm outperforms the naive-nearest algorithm in 2/3 scenarios, supporting the theory that response

accuracy can be improved by leveraging the inherent analogical structure of the embedding space.

Surprisingly, the same pattern was not observed when comparing the flow vectors and ANN classifier algorithms. Like naive-analogy and naive-nearest, these two algorithms differ primarily in their use of analogical structure. The fact that ANN tends to outperform flow vectors suggests that in this case, the averaging of multiple dialog trajectories (a.k.a. "flow vectors") results in a target point that lies far from the manifold of valid responses, whereas the averaging of multiple actual sentence embeddings (as per ANN) remains closer to the valid manifold. Further research is needed to understand this phenomenon and quantify the complex structures found in semantic embedding spaces, and in this direction we applaud the work of [31] and [11].

3.2 Generalization Across Datasets

We found ourselves curious as to what extent each dialog corpus was able to generalize to the other corpora in the evaluation set. We therefore implemented the following experiment: Using the scattershot algorithm and embedded concatenation localization method, we created a confusion matrix showing how well the algorithm performed when using only one of our four corpora as its scaffold.

Table 3. Confusion matrix showing how well each dataset, when used as a scaffold corpus, is able to select appropriate responses for dialogs drawn from the other corpora. Each column contains a scaffold corpus, each row an evaluation corpus. The scattershot algorithm was used in conjunction with the embedded concatenation localization method, with a context size $n = 2$ and with 3,311 evaluation dialogs drawn from each corpus. The highest accuracy level in each column is shown in bold-face text.

	Chit-Chat	Daily Dialog	Reddit	Ubuntu
Chit-Chat	.5826	**.6563**	.6068	.3455
Daily Dialog	.5421	.6540	.5847	.3582
Reddit	**.5886**	.6261	**.7508**	.3860
Ubuntu	.4971	.4666	.5639	**.7523**

Results are shown in Table 3. The unusually high performance seen when both the scaffold and evaluation corpus were taken from the Ubuntu dataset can be explained by the high level of overlap within the Ubuntu dialog corpus. An investigation of the data downloaded from Kaggle reveals that between the three files ("dialogueText_301.csv", "dialogueText_196.csv", and "dialogueText.csv") there was an overlap of 53.35% in the original data (14,318,055 non-unique turns out of a total of 26,839,031 turns). As a result, the evaluation corpus drawn from the Ubuntu dataset contained exact copies of dialogs in the Ubuntu scaffold corpus. The other corpora had little or no overlap.

3.3 Retrieval from Large Data Repositories

The end objective of our research is to facilitate the creation of versatile conversational systems that are able to select dialog responses the conform to a friendly, upbeat and courteous conversational style. Candidate utterances for such systems may be generated by an ensemble of response generators, retrieved from a databank of possible utterances, or created algorithmically via a context-free grammar or template system, but in all cases there must be a means of deciding which candidate utterance is "best", i.e., which one should be delivered to the user. In generative systems, too, it is critical that we develop more and better methods for conversational scoring. Metrics such as n-gram matching, METEOR [3], BERTScore [29] and perplexity are helpful, but not yet sufficient for the aspirations of the research community.

To explore the effectiveness with which our system ranks responses and the types of conversations it is able to produce, we allowed a human to type the opening line of dialog with an agent executing the scattershot algorithm. The objective was to see whether a dialog agent choosing from among ca. 10,000 candidate responses could leverage the analogical structure of language in order to create a plausible conversation. A randomly-selected sentence from a repository this size would, of course, be meaningless and generally widely off-topic. We wanted to see if our system could do better.

Figures 4 and 5 show typical conversations generated using the method described above. Interestingly, although the 19,665 randomly-selected candidate utterances were drawn from all four conversational datasets, almost all of the ones chosen by the scattershot algorithm came from the Chit-Chat dataset. This suggests that the Chit-Chat dataset was an unusually good stylistic match for the informal conversation patterns used by the human chatter.

We permitted one augmentation to our algorithms for this experiment: Candidate responses that were too similar to the most recent statement in the dialog history were excluded from consideration[6]. This constitutes an extension at the sentence level of the traditional exclusion of source words when solving analogical queries via word embeddings [14,20]. Without it, the scaffolding algorithm tends to select sentences that parrot or reflect the content of the dialog history rather than progressing to new topics.

4 Analysis and Future Work

A key insight to be taken from our results is the observation that the average of several valid responses cannot be assumed to *also* represent a valid response. This amounts to taking the centroid of a (possibly oddly-shaped) manifold, resulting in a vector that might easily represent a nonsense sentence. For example, it is important to realize that the 'ideal' responses d_i depicted in Fig. 3 do not necessarily represent an optimal reply; they may in fact be the vector embedding

[6] Similarity was defined as Euclidean distance $< \tau$, where τ is a hand-selected threshold value.

```
> Hey. What's up?
Hey, whats up?
> Not much. How is your day going so far?
Hi, it's good!
> Good to hear. What are you working on?
I'm figuring out my budget .
> Oh that sounds fun. What are you going to do after you finish?
Well I plan to get into the accounting program
> Oh cool! Do you like accounting?
I've thought about going into cardiology, that seems interesting, but also I'm
considering anesthesiology.
> Oh wow. Those are both very different from accounting. Are they hard?
semester for you and me. Is your next semester seeming to look like its going to
be a little easier?
> I think this semester will be fun but a little challenging.
That it does! Even though you may not be getting everything out of your classes
now but oh well...;)
> Do you think you will have any time to go to see a movie this semester?
That it does! Even though you may not be getting everything out of your classes
now but oh well...;)
> So, where are you from originally?
Haha well I'm from Arkansas
> Oh cool. I have never been there. Do you like it?
Ya actually I did enjoy the shreveport one when I lived there last year (2012
Mardi gras) but you're right, it's almost nothing Compared to most of the NO ones.
Still very enjoyable. My of my friends lives right on the route so it was nice
having somewhere to go for it lol
> haha fun. Do you like it here then?
Cool
> If you could vacation anywhere, where would it be?
well,its like 1 am here,i should get to bed, it was nice talking to you, have a
great day.
> Oh wow. Goodnight. Talk to you later.
Thanks. You too!
```

Fig. 4. Conversation generated using the scattershot scaffolding algorithm, the embedded concatenation localization method, and distance threshhold $\tau = 0.6$. Human-generated text is preceded by the $>$ symbol. In this example, even the τ threshhold is not sufficient to keep the agent from getting caught in a sentence repetition, however, it successfully switches to a new topic on the next utterance. Conversation originally published in [27].

of a nonsense sentence. Rather, these 'ideal' responses, or target points, function as a heuristic to gauge whether a specific candidate response is conversationally appropriate.

A future direction for this research should include the use of a sufficiently-trained decoder to render the target point directly into text, thus enabling researchers to probe the structure and behavior of the embedding space in more detail.

```
> Hello.
Hey!
> How are you doing today?
How are you?
> I am good. How has your day been so far?
I've been pretty good, I was sick earlier, but better now.
> Oh man, that stinks. What were you sick with? I am glad you are feeling better
though!
I had like ten servings of it.
> Uh oh. That sounds like a lot. Was it good food?
I had such tasty food too.
> Yum. What is your favorite food?
yum sounds so good right now haha! I love every fruit! I like rice and pickles too
haha. but like you I like just about everything
> I love fruit too. So, have you seen any good movies lately?
Black panther is so good!!!
> Yeah I really liked it too. I think that or, Thor Ragnarok might be my favorite
Marvel movie.
No , I haven't . This is my first time to watch it .
> Oh yeah?
*It flips him off.*
> What are some of your favorite books?
It's a little bit of romance and comedy.
> Cool. Well, I have to go. It has been nice talking to you!
Okay ! Bye !
```

Fig. 5. Conversation generated using the scattershot scaffolding algorithm and Euclidean distance localization method, and distance threshold $\tau = 0.5$. Human-generated text is preceded by the > symbol. In this example, the agent is able to maintain several fairly coherent dialog turns, then pivots appropriately to the topic of movies in response to user cues. It also successfully detects and responds to an indication that the conversation is over. Conversation originally published in [27].

Another useful direction for future research would be the use of a dynamic context length depending on the content of the dialog history. For example, generic sentences such as "yes", "of course not", or "i'm not sure" provide little conversational context, and are generally meaningless in isolation, whereas other sentences may require little or no context at all in order to enable an optimal response. The taks of dynamically determining when further context is needed, and how much of it to include, is a fertile area for future research.

Going forward, we imagine a possible future agent which generates responses via a neural architecture, but which has been trained to adhere as closely as possible to a scaffold corpus in its utterance patterns. Future work in this area should explore neural dialog models that utilize a scaffold corpus during loss calculations. A comprehensive study of distance metrics should also be undertaken, as it is not necessarily certain that the *de facto* standards of Euclidean and cosine distance are the best possible heuristics for semantic similarity; L1 distance or correlation coefficients might be more effective.

As new language models and embedding algorithms are constantly being developed, another important area for future work involves the testing and analysis of new semantic embedding spaces, with an eye towards identifying the ones that are most appropriate for conversational scaffolding, analogical reasoning, and other processes that depend on the semantic properties and innate geometry of the embedding space itself.

5 Conclusion

As automated personal assistants become more prevalent, developers will need to strike a balance between control and spontaneity. We want our conversational systems to behave in unexpected, even surprising ways, even pushing humans out of their comfort zone at times. But we also want them to be kind and courteous, and refrain from insulting their users, making broadly offensive statements, or giving inaccurate information. Striking this balance requires *finesse*, and we believe that conversational scaffolding strikes a good balance between leveraging the power of connectionist systems and maintaining the continuity of a heavily curated system.

The methods outlined in this paper describe an enticing middle ground, allowing a scaffold corpus to define an overall personality or conversational style for the agent without directly restricting its responses. In this paper, we have presented a scaffolding algorithm that uses pre-trained sentence embeddings to (a) leverage the inherent analogical properties of the embedding space and (b) account for the one-to-many property of language while (c) encouraging responses that closely align with the scaffold corpus. Our scattershot algorithm is able to predict the correct follow-on sentence for a given dialog history with nearly 70% accuracy, outperforming both ANN and naive nearest-neighbor baselines. It is also able to produce engaging and (sometimes) believable conversations with topical coherence a relaxed, conversational feel. We believe that conversational scaffolding and the scattershot algorithm offer a unique and valuable new paradigm for response ranking in open-domain dialog settings, and we look forward to future work in this area.

Acknowledgements. We wish to thank David Wingate and his students in the BYU Perception, Control and Cognition laboratory for their role in creating and hosting the Chit-Chat dataset, and Daniel Ricks for his contributions to Fig. 3.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015). https://www.tensorflow.org/
2. Bak, J., Oh, A.: Variational hierarchical user-based conversation model. 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), pp. 1941–1950. Association for Computational Linguistics, Hong Kong, China, November 2019. https://doi.org/10.18653/v1/D19-1202, https://www.aclweb.org/anthology/D19-1202

3. Banerjee, S., Lavie, A.: Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72 (2005)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
5. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. CoRR abs/1511.06349 (2015). http://arxiv.org/abs/1511.06349
6. Buckels, E.E., Trapnell, P.D., Paulhus, D.L.: Trolls just want to have fun. Pers. Individ. Differ. **67**, 97–102 (2014)
7. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: semantic textual similarity - multilingual and crosslingual focused evaluation. In: Proceedings of SemEval, vol. 2017 (2017)
8. Cer, D., et al.: Universal sentence encoder. CoRR abs/1803.11175 (2018). http://arxiv.org/abs/1803.11175
9. Cho, D., Acquisti, A.: The more social cues, the less trolling? an empirical study of online commenting behavior (2013)
10. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 (2017)
11. Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single vector: Probing sentence embeddings for linguistic properties. arXiv preprint arXiv:1805.01070 (2018)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
13. Fulda, N., et al.: Byu-eve: mixed initiative dialog via structured knowledge graph traversal and conversational scaffolding. In: Proceedings of the 2018 Amazon Alexa Prize, November 2018
14. Gladkova, A., Drozd, A., Matsuoka, S.: Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In: Proceedings of the NAACL Student Research Workshop, pp. 8–15 (2016)
15. Kiros, R., et al.: Skip-thought vectors. CoRR abs/1506.06726 (2015)
16. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. arXiv e-prints arXiv:1710.03957 (2017)
17. Logeswaran, L., Lee, H.: An efficient framework for learning sentence representations. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=rJvJXZb0W
18. Lowe, R., Pow, N., Serban, I., Pineau, J.: The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. arXiv e-prints arXiv:1506.08909 (2015)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
20. Mikolov, T., tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. Association for Computational Linguistics, May 2013

21. Park, Y., Cho, J., Kim, G.: A hierarchical latent structure for variational conversation modeling. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers), vol. 1, pp. 1792–1801. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. https://doi.org/10.18653/v1/N18-1162, https://www.aclweb.org/anthology/N18-1162
22. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). http://www.aclweb.org/anthology/D14-1162
23. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
24. Rainie, H., Anderson, J.Q.: The future of free speech, trolls, anonymity and fake news online (2017)
25. Reddit: Reddit datasets. https://www.reddit.com/r/datasets/
26. Shen, X., Su, H., Niu, S., Demberg, V.: Improving variational encoder-decoders in dialogue generation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
27. Will, M., Tyler, E., Nancy, F.: Conversational scaffolding: an analogy-based approach to response prioritization in open-domain dialogs. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART) (2020)
28. Xupeng Tong, Y.L., Yen, C.M.: Variational neural conversational model. In: ICML (2014). https://www.cs.cmu.edu/epxing/Class/10708-17/project-reports/project12.pdf
29. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
30. Zhao, T., Zhao, R., Eskenazi, M.: Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers), vol. 1, pp. 654–664. Association for Computational Linguistics, Vancouver, Canada, July 2017. https://doi.org/10.18653/v1/P17-1061, https://www.aclweb.org/anthology/P17-1061
31. Zhu, X., Li, T., De Melo, G.: Exploring semantic properties of sentence embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), vol. 2, pp. 632–637 (2018)
32. Zhu, Y., et al.: 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, pp. 19–27 (2015)