Alignment over Heterogeneous Embeddings for Question Answering

Vikas Yadav, Steven Bethard, Mihai Surdeanu
University of Arizona, Tucson, AZ, USA
{vikasy,bethard,msurdeanu}@email.arizona.edu

Abstract

We propose a simple, fast, and mostly-unsupervised approach for non-factoid question answering (QA) called Alignment over Heterogeneous Embeddings (AHE). AHE simply aligns each word in the question and candidate answer with the most similar word in the retrieved supporting paragraph, and weighs each alignment score with the inverse document frequency of the corresponding question/answer term. AHE's similarity function operates over embeddings that model the underlying text at different levels of abstraction: character (FLAIR), word (BERT and GloVe), and sentence (InferSent), where the latter is the only supervised component. Despite its simplicity and lack of supervision, AHE obtains a new state-of-the-art performance on the "Easy" partition of the AI2 Reasoning Challenge (ARC) dataset (64.6% accuracy), top-two performance on the "Challenge" partition of ARC (34.1%), and top-three performance on the WikiQA dataset (74.08% MRR), outperforming many other complex, supervised approaches. Our error analysis indicates that alignments over character, word, and sentence embeddings capture substantially different semantic information. We exploit this with a simple meta-classifier that learns how much to trust the predictions over each representation, which further improves the performance of unsupervised AHE[1].

1 Introduction

The "deep learning tsunami"(Manning, 2015) has had a major impact on important natural language processing (NLP) applications such as question answering (QA). Many neural approaches for QA have been proposed in the past few years, with impressive results on several QA tasks (Seo et al., 2016; Wang and Jiang, 2016; Wang et al., 2017b; Tymoshenko et al., 2017; Xiong et al., 2016a; Wang et al., 2018; Radford et al., 2018; Li et al., 2018, inter alia). However, an undesired effect of this focus on neural approaches was that other methods have fallen out of focus, including strong unsupervised benchmarks that are necessary to highlight the true gains of supervised approaches. For instance, alignment approaches have received considerably less interest recently, despite their initial successes (Echihabi and Marcu, 2003; Surdeanu et al., 2011, inter alia). While a few recent efforts have adapted these alignment methods to operate over word representations (Kenter and De Rijke, 2015; Kim et al., 2017; Yadav et al., 2018), they generally underperfom supervised neural methods due to their underlying bag-of-word (BoW) assumptions and reliance on uncontextualized word representations such as GloVe (Pennington et al., 2014).

Question - Which sequence of energy transformations occurs after a battery-operated flashlight is turned on?

1. *electrical → light → chemical*
2. *electrical →chemical → light*
3. *chemical → light → electrical*
4. **chemical → electrical → light**

Supporting paragraph(s): "a chemical cell converts chemical energy into electrical energy; a flashlight chemical energy to light energy"

Figure 1: A multiple-choice question from the ARC dataset with the correct answer in bold font. This question is answered correctly by our alignment method that relies on contextualized word embeddings that capture the correct sequence, and cannot be answered correctly when relying on uncontextualized embeddings.

In this work we argue that alignment approaches are more meaningful today after the advent of *contextualized* word representations, which mitigate the above BoW limitations. For example, Figure 1 shows an example of a question from AI2's Reasoning Challenge (ARC) dataset (Clark et al., 2018),

[1]Code: https://github.com/vikas95/AHE

Proceedings of NAACL-HLT 2019, pages 2681–2691
Minneapolis, Minnesota, June 2 - June 7, 2019.

which is not answered correctly by a state-of-the-art BoW alignment method (Yadav et al., 2018), but is correctly answered by our alignment approach when operating over Bidirectional Encoder Representations from Transformers (BERT) embeddings (Devlin et al., 2018).

We propose a simple, fast, and mostly-unsupervised approach for non-factoid QA called Alignment over Heterogeneous Embeddings (AHE). AHE uses an off-the-shelf information retrieval (IR) component to retrieve likely supporting paragraphs from a knowledge base (KB) given a question and candidate answer. Then AHE aligns each word in the question and candidate answer with the most similar word in the retrieved supporting paragraph, and weighs each alignment score with the inverse document frequency (IDF) of the corresponding question/answer term. AHE's overall alignment score is the sum of the IDF weighted scores of each of the question/answer term.

Importantly, AHE's alignment function operates over contextualized embeddings that model the underlying text at different levels of abstraction: character (FLAIR) (Akbik et al., 2018), word (BERT) (Devlin et al., 2018), and sentence (InferSent) (Conneau et al., 2017), where the latter is the only supervised component in the proposed approach. The different representations are combined through an ensemble approach that by default is unsupervised (using a variant of the NoisyOr formula), but can be replaced with a supervised meta-classifier.

The contributions of our work are the following:

1. To our knowledge, this is the first unsupervised alignment approach for QA that: (a) operates over contextualized embeddings, and (b) captures text at multiple levels of abstraction, including character, word, and sentence.
2. We obtain (near) state-of-the-art results (top three or higher) on three QA datasets: WikiQA (Yang et al., 2015) (74.08 mean reciprocal rank), ARC the Challenge partition (34.1% precision at 1 (P@1)) and ARC Easy (64.6 P@1). Our approach outperforms information retrieval methods, other unsupervised alignment approaches, and many supervised, neural approaches, despite the fact that it is mostly unsupervised and much simpler. Importantly, unlike many neural approaches, our results are robust across several datasets. Minimally, these results indicate that the work proposed here should be considered as a new, strong baseline for the task.
3. Our analysis indicates that alignments over character, word, and sentence embeddings capture substantially different semantic information. We highlight this complementarity with an oracle system that chooses the correct answer when it is proposed by any of the AHE's representations, which achieves 68% P@1 on ARC Challenge, 86% on ARC Easy, and 93.7% mean average precision (MAP) on WikiQA. We exploit this complementarity with a simple meta-classifier that learns when and how much to trust the predictions over each representation, which further improves the performance of unsupervised AHE.

2 Related Work

We highlight major trends in the field, and how our work compares with them. We focus mostly on non-factoid QA, which is usually implemented in two forms: multiple-choice QA such as AI2's Reasoning Challenge (ARC), where the answer must be selected from multiple candidates and (optionally) supported by explanatory texts extracted from external knowledge bases (Clark et al., 2018); or answer sentence selection, where candidate answer sentences are provided and the task is to select the sentences containing the correct answers (Yang et al., 2015). Alignment models have also been proposed for other types of QA, such as reading comprehension (RC) QA (Chakravarti et al., 2017). We believe AHE can be similarly extended to RC, but, in this work, we have limited our experiments to answer selection and multiple-choice QA tasks.

Most QA approaches today use neural, supervised methods. Most use stacked architectures usually coupled with attention mechanisms (He and Lin, 2016; Yin et al., 2015; Seo et al., 2016; Xiong et al., 2016b; Kumar et al., 2016; Tan et al., 2015; Wang et al., 2017a; Chen et al., 2016; Cheng et al., 2016; Golub and He, 2016). Some of these works also rely on structured knowledge bases (Zhong et al., 2018a; Ni et al., 2018) such as ConceptNet (Speer et al., 2017). Some approaches use query expansion methods in addition to the above methods (Musa et al., 2018; Nogueira and Cho, 2017; Ni et al., 2018). For example, Musa et al. (2018) used a sequence to sequence model (Sutskever et al., 2014) to generate an enhanced query for ARC which retrieves better supporting passages.

However, in general, all these approaches rely

on annotated training data, and, some, on structured KBs, which are expensive to create (Jauhar et al., 2016). Further, as we demonstrate in Section 5, these methods tend to be tailored to a specific dataset and do not port well to other domains or even within different splits of the same dataset. In contrast, our method is mostly unsupervised and does not require training. Even then, our approach performs well on three distinct QA datasets, with top three performance in all.

Our work is inspired by previous efforts on using alignment methods for NLP (Echihabi and Marcu, 2003). Unsupervised alignment models have been proposed for several NLP tasks such as short text similarity (Kenter and De Rijke, 2015), answer phrase/sentence selection in reading comprehension (RC) (Chakravarti et al., 2017), document retrieval (Kim et al., 2017), etc. Other works have utilized word alignments as features in supervised models (Surdeanu et al., 2011; Wang and Ittycheriah, 2015). For example, Wang and Ittycheriah (2015) utilized the alignment of words between two questions as a feature in a feedforward neural network that matches similar FAQ questions. Recently, Yadav et al. (2018) showed that alignment methods remain competitive for non-factoid QA.

However, the majority of alignment models that rely on representation learning utilize uncontextualized word embeddings such as GloVe, coupled with other BoW models such as IBM Model 1 (Brown et al., 1993) for alignment (Kenter and De Rijke, 2015; Kim et al., 2017; Yadav et al., 2018). To our knowledge, we are the first to adapt these ideas to contextualized embeddings, which mitigates the BoW limitations of previous efforts (as shown in Figure 1). While contextualized representations have been shown to be extremely useful for multiple NLP tasks (Devlin et al., 2018; Peters et al., 2018; Howard and Ruder, 2018), our work is the first to apply them to an unsupervised alignment approach. Further, we show that different contextualized representations of text (character, word, sentence) capture complementary information, and combining them improves performance further.

3 Approach

The core component of our approach computes the score of a candidate answer by aligning two texts. For multiple-choice questions, the first text consists of the question concatenated with the candidate answer, and the second is a supporting paragraph such as the one shown in Figure 1, which consists of one or more sentences retrieved from a larger textual KB using an off-the-shelf IR system (Section 3.1). For answer selection tasks, the first text is the question and the second is the sentence that contains the candidate answer. Answer candidates are then sorted in descending order of their alignment scores. In both cases, the alignment approach operates over multiple contextualized embeddings that model the two texts at different levels of abstraction: character, word, and sentence. The overall architecture is illustrated in Figure 2. We detail the alignment method in §3.2, the multiple representations of text considered in §3.3, and the ensemble strategies over these representations in §3.4.

3.1 Retrieving Supporting Paragraphs

For multiple-choice question datasets such as ARC, we retrieve supporting information from external KBs using Lucene, an off-the-shelf IR system[2]. We use as query the question concatenated with the corresponding answer candidate, and BM25 (Robertson et al., 2009) as the ranking function[3]. For each query, we keep the top C Lucene documents, where each document consists of a sentence retrieved from the ARC corpus. Similar to our previous work (Yadav et al., 2018), we boost candidate answer terms by a factor of 3 while keeping question terms as it is in the BM25 ranking function. All texts were preprocessed by discarding the case of the tokens, removing the stop words from Lucene's list, and lemmatizing the remaining tokens using NLTK (Bird, 2006). For all experiments reported on the ARC dataset we used $C = 20$.

Here we also calculate the IDF of each query term q_i (required later during alignment):

$$idf(q_i) = \log \frac{N - docfreq(q_i) + 0.5}{docfreq(q_i) + 0.5} \quad (1)$$

where N is the number of documents (e.g., 14.3M for the ARC KB) and $docfreq(q_i)$ is the number of documents that contain q_i.

3.2 Alignment Algorithm

For representations that produce word embeddings (e.g., FLAIR, BERT, GloVe), we use the alignment algorithm in Figure 3. Our method computes the alignment score of each query token with every token in the given KB paragraph, using the cosine

[2] https://lucene.apache.org

[3] https://lucene.apache.org/core/7_0_1/core/org/apache/lucene/search/similarities/BM25Similarity.html

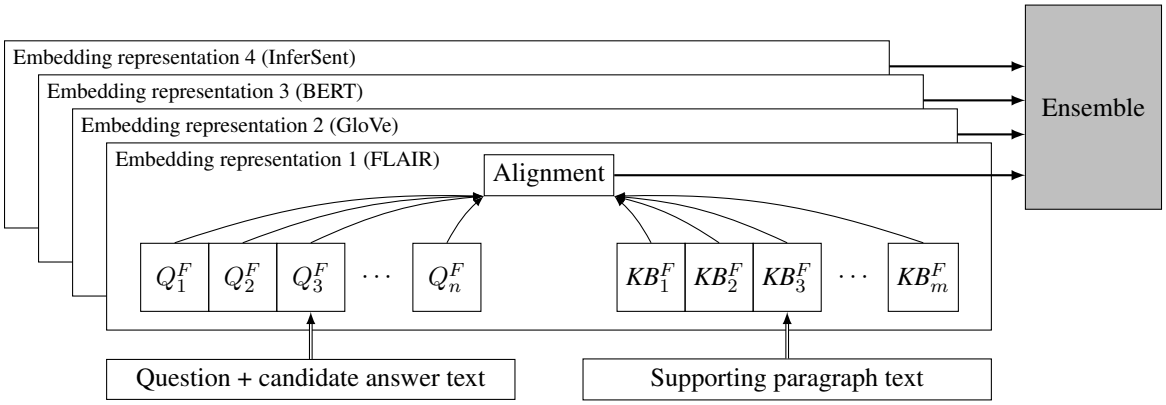


Figure 2: AHE architecture illustrated for the multiple-choice question setting. The left text consists with of the question concatenated with the answer candidate; the right text is a supporting paragraph retrieved from an external KB. The same alignment score is computed over multiple representations of text, and then aggregated through an ensemble model.

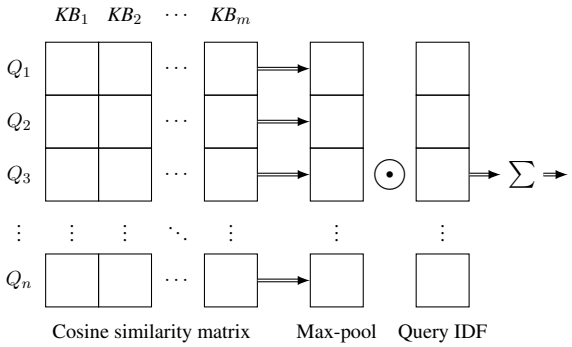


Figure 3: Alignment component of AHE, where a cosine similarity matrix is constructed by comparing token embeddings of input query tokens (Q_i) and supporting KB paragraph tokens (KB_j), and the maximal alignment cosine score for each input query token is weighted by its IDF.

similarity of the two embedding vectors. Then, a max-pooling layer over this cosine similarity matrix is used to retrieve the most similar token in the supporting passage for each query token. Lastly, this max-pooled vector of similarity scores is multiplied with the vector containing the IDF values of the query tokens and the resultant vector is summed to produce the overall alignment score s for the given query Q_a (formed from question Q and candidate answer a) and the supporting paragraph P_j:

$$s(Q_a, P_j) = \sum_{i=1}^{|Q_a|} idf(q_i) \cdot align(q_i, P_j) \quad (2)$$

$$align(q_i, P_j) = \max_{k=1}^{|P_j|} cosSim(q_i, p_k) \quad (3)$$

$$cosSim(q_i, p_k) = \frac{\vec{q_i} \cdot \vec{p_k}}{||\vec{q_i}|| \cdot ||\vec{p_k}||} \quad (4)$$

where $\vec{q_i}$ and $\vec{p_k}$ are the embedding vectors of the terms q_i and p_k.

In addition to alignments over word-level embeddings, we include InferSent (Conneau et al., 2017), which generates sentence-level embeddings (see §3.3 for details). For InferSent, the alignment score between a query Q_a and a supporting paragraph P_j is computed as the dot product of the two corresponding sentence vectors, $\vec{Q_a}$ and $\vec{P_j}$, normalized using softmax over all candidate answers:

$$s(Q_a, P_j) = softmax(\vec{Q_a} \cdot \vec{P_j}) \quad (5)$$

For ARC, the above alignment scores are computed for each supporting paragraph in the set of C paragraphs retrieved in §3.1. For WikiQA, this score is computed just for the sentence containing the candidate answer.

To aggregate the retrieved ARC paragraph scores (for ARC) into an overall score for the corresponding candidate answer, we consider:

Max: selects the maximum alignment score between all available paragraphs as the final score for candidate answer a:

$$S(cand_a) = \max_{j=1}^{C}(s(Q_a, P_j)) \quad (6)$$

Weighted average: averages all available paragraph scores, using as weights the inverse IR ranks of the corresponding paragraphs:

$$S(cand_a) = \sum_{j=1}^{C} \frac{1}{j}(s(Q_a, P_j)) \quad (7)$$

During tuning, we observed that the max strategy is better for ARC Challenge, while the weighted average is better for ARC Easy. We conjecture that this happens because Challenge questions require

information that is sparser in the collection, and, thus, including more than the top paragraph tends to introduce noise.

3.3 Text Representations

AHE computes alignments over four different embedding representations that model the text at different levels of abstraction: character, word, and sentence (as detailed below). Although all these embeddings can be tuned for specific domains to improve performance, here we highlight the potential of publicly-available, pre-trained embeddings. Hence, we did not train embeddings on any domain specific corpus, and directly used off-the-shelf embeddings in all but one situation. The details of all four component embeddings of AHE are discussed below.

Character-based embeddings: We used the FLAIR contextual character language model of Akbik et al. (2018). They used long short-term memory (LSTM) networks that operate at character level over the entire text to generate character embeddings (in both forward and backward directions). Similar to them, to generate the embedding for token i, we concatenate the embedding from the forward LSTM for the character following the token, with the embedding from the backward LSTM for the character preceding the token:

$$\mathbf{w}_i^{FLAIR} := \begin{bmatrix} \mathbf{h}^f_{t_i+1-1} \\ \mathbf{h}^b_{t_i-1} \end{bmatrix} \quad (8)$$

where t_i is the character offset of the i^{th} token in the input text, and $\mathbf{h}$ is the corresponding LSTM's hidden state. We used the "mix-forward" and "mix-backward" pretrained models provided by the authors to produce two character embeddings, each of size 2048, resulting in word embeddings of size 4096.

Word-based embeddings: We incorporated two different word-based embeddings:
BERT – we used the Bidirectional Encoder Representations from Transformers (BERT) embedding model of Devlin et al. (2018). We concatenated the last four layers (as suggested by the authors[4]) of the BERT Large language model, where each layer has size 1024, summing up to size 4096 embeddings for each token:

$$\mathbf{w}_i^{BERT} := [Layer_{-1}, ..., Layer_{-4}] \quad (9)$$

[4] https://github.com/google-research/bert

GloVe – we also include GloVe embeddings (Pennington et al., 2014), under the hypothesis that these uncontextualized word embeddings will provide complementary information to the contextualized BERT embeddings. We used GloVe embeddings of size 300, trained over 840B tokens from Wikipedia, resulting in 2.2M words vocabulary.

Sentence-based embeddings: Lastly, we used InferSent, the sentence-based embeddings of Conneau et al. (2017). InferSent was originally trained on several natural language inference (NLI) datasets to generate the sentence representations that maximize the probability of correct inference. This model achieved poor performance on our QA tasks (see rows 8a in Table 1 and row 7a in Table 2).

Therefore, rather than using this NLI model, we trained InferSent on our data by maximizing the inference probability from the input query[5] to the supporting paragraph. We used the same number of supporting passages ($C = 20$) and the same scoring functions as explained in Section 3.2. We trained InferSent using batches of size 32, the Adam optimizer, learning rate = 0.001, and 50 epochs. We used max pooling over the token's LSTM hidden states to generate an overall sentence embedding. We tuned the sentence representation size on the development sets,[6] which resulted in 128 for WikiQA and 384 for ARC.

3.4 Aggregating Multiple Representations

We aggregate the scores of candidate answers over the four different embedding representations using an unsupervised variant of the NoisyOr formula:

$$NoisyOr_M(i) = 1 - (\prod_{m=0}^{M} (1 - \alpha^m * S_i^m)) \quad (10)$$

which computes the overall score for answer candidate i. M is the total number of representations (e.g., 4 in our case), and S_i^m is the score of answer candidate i under representation m. Lastly, α^m is a hyperparameter used to dampen peaky distributions of answer probabilities. We included this hyperparameter because we observed that InferSent produces a probability distribution over candidate answers where one answer tends to take most of the probability mass, and these scores dominate in the NoisyOr. Thus, the α^m weights are set to 1 for

[5] In ARC, the input query concatenates the question with a candidate answer.

[6] This was a light process that inspected only five possible values: 64, 128, 256, 384, and 512.

#	Supervised?	Type of KB	Model	Easy P@1	Challenge P@1
			Baselines		
1	No	text	Random baseline	25.02	25.02
2	No	text	AI2 IR Solver (Clark et al., 2018)	59.99	23.98
3	No	text	AI2 IR Solver (our implementation)	60.31	23.74
4	No	text	Sanity Check (Yadav et al., 2018)	58.36	26.56
5	No	text	AHE over GloVe	60.71	28.75
6	No	text	AHE over FLAIR	62.29	31.05
7	No	text	AHE over BERT	62.73	32.76
8a	No	text	AHE over InferSent (trained on NLI)	32.13	25.36
8b	Yes	text	AHE over *InferSent* (trained on ARC)	54.01	31.66
			Previous work		
9	Yes	text, structured	Tuple-Inf (Clark et al., 2018)	60.71	23.83
10	Yes	text	Decomp-att (Clark et al., 2018)	52.95	24.40
11	Yes	text, structured	DGEM (Clark et al., 2018)	58.97	27.11
12	Yes	text	BiDAF (for ARC) (Clark et al., 2018)	51.05	26.54
13	Yes	text, structured	KG2 (Zhang et al., 2018)	-	31.70
14	Yes	–	Bi-LSTM max-out (Mihaylov et al., 2018)	34.26	33.87
15	Yes	text, structured	NCRF++/match-LSTM (Musa et al., 2018)	52.22	33.20
16	Yes	text, structured	TriAN+f(dir)(cs)+f(ind)(cs) (Zhong et al., 2018b)	-	33.39
17	Yes	text, structured	ET-RR (Ni et al., 2018)	-	**36.56**
			Unsupervised AHE		
18	No	text	AHE (FLAIR+BERT)	63.45	33.87
19	No	text	AHE (FLAIR+BERT+GloVe)	64.60	31.06
20	Minimal	text	AHE (FLAIR+BERT+*InferSent*)	62.21	34.05
21	Minimal	text	AHE (FLAIR+BERT+GloVe+*InferSent*)	63.22	33.28
			Supervised AHE		
22	Yes	text	*AHE* (FLAIR+BERT+GloVe+*InferSent*)	65.19	33.70
23	Yes	text	*AHE* (FLAIR+BERT+GloVe+*InferSent*) with grade	**65.66**	34.47
			Oracle		
24	–	text	Oracle ensemble (FLAIR+BERT+GloVe+*InferSent*)	85.11	68.09

Table 1: Performance on the ARC dataset, measured using precision at 1 (P@1), on both the Easy and Challenge partitions. Italic font indicates which AHE components are supervised, e.g., *InferSent* is the InferSent model trained on ARC data; *AHE* is the AHE variant that uses the supervised meta-classifier ensemble. Line 8a shows performance of alignment over the original InferSent embeddings (trained on NLI datasets); line 8b shows performance when using *InferSent* embeddings trained on ARC training data. The "minimal" supervision configurations (lines 20 and 21) include the supervised *InferSent*, but use the unsupervised NoisyOr strategy for aggregation.

all representations with the exception of InferSent, for which we tuned its value to 0.2.

Of course, other types of aggregation are possible. To explore this space, we also implemented a supervised meta-classifier, which aims to learn the aggregation function directly from data. We implemented this multi-classifier as a feed forward network with two fully connected dense layers of hidden size 16 and K respectively, where K is the maximum number of candidate answers for the given dataset. The activation function of the first dense layer was tanh; we used a softmax in the second output layer. The input to this network was a vector of size $M \times K$. For example, for ARC this vector has a size $4 \times 5 = 20$. For WikiQA this vector has size $4 \times 22 = 88$. Each element in the input vector is the score of one candidate answer under a given representation. Additionally, for ARC we used an extra position in the input vector to indicate the grade of the corresponding exam question (provided in the dataset) with the intuition that the meta-classifier will learn to trust different representations for different grade levels.

4 Empirical Results

We evaluate AHE on two QA tasks:

AI2's Reasoning Challenge (ARC): this is a multiple-choice question dataset, containing science exam questions (Clark et al., 2018). The dataset is split in two partitions: Easy and Challenge, where the latter partition contains the more difficult questions that require reasoning. Each partition is split into train/development/test as follows: Easy contains 2251/570/2376 questions, and Challenge 1119/299/1172. Most of the questions have 4 answer choices, with only $< 1\%$ of all the questions having either 3 or 5 answer choices. ARC also includes a textual KB of 14.3M passages suitable for solving ARC questions. Note that we use solely this KB for retrieving supporting paragraphs,

#	Supervised?	Model	MAP	MRR
		Baselines		
1	No	Wgt Word Cnt (Yang et al., 2015)	50.99	51.32
2	Yes	LCLR (Yang et al., 2015)	59.93	60.86
3	No	Sanity Check (Yadav et al., 2018)	64.02	-
4	No	AHE over GloVe	63.40	65.39
5	No	AHE over FLAIR	64.91	66.51
6	No	AHE over BERT	65.13	66.40
7	Yes	AHE over *InferSent*	66.93	68.70
		Previous work		
8	Yes	CNN+Cnt (Yang et al., 2015)	65.20	66.52
9	Yes	RNN-1way (Jurczyk et al., 2016)	66.64	68.70
10	Yes	RNN-Attention_pool (Jurczyk et al., 2016)	67.47	68.92
11	Yes	CNN (avg + emb) (Jurczyk et al., 2016)	68.78	70.82
12	Yes	AP-CNN (dos Santos et al., 2016)	68.86	69.57
13	Yes	LSTM-att (Miao et al., 2016)	68.86	70.69
14	Yes	ABCNN (Yin et al., 2015)	69.21	71.08
15	Yes	Key-value memory network (Miller et al., 2016)	70.69	72.65
16	Yes	CubeCNN (He and Lin, 2016)	70.90	72.34
17	Yes	BiMPM (Wang et al., 2017b)	71.80	73.10
18	Yes	(Tymoshenko et al., 2017)	72.19	74.08
19	Yes	Compare-Aggregate (Wang and Jiang, 2016)	74.33	75.45
20	Yes	(Li et al., 2018)	**75.41**	**76.59**
		Unsupervised AHE		
21	No	AHE (FLAIR+BERT)	66.98	68.10
22	No	AHE (FLAIR+BERT+Glove)	67.31	68.53
23	Minimal	AHE (FLAIR+BERT+*InferSent*)	71.52	73.85
24	Minimal	AHE (FLAIR+BERT+Glove+*InferSent*)	71.77	74.08
		Supervised AHE		
25	Yes	*AHE* (FLAIR+BERT+Glove+*InferSent*)	72.13	74.64
		Oracle		
26	–	Oracle ensemble (FLAIR+BERT+Glove+*InferSent*)	93.71	95.49

Table 2: Performance on the WikiQA dataset, measured using mean average precision (MAP) and mean reciprocal rank (MRR). Italic font indicates which AHE components are supervised, e.g., *InferSent* is the InferSent model trained on WikiQA data; *AHE* is the AHE variant that uses the supervised meta-classifier ensemble. The "minimal" supervision configurations (lines 23 and 24) include the supervised *InferSent*, but use the unsupervised NoisyOr strategy for aggregation.

unlike many other approaches that use additional structured KBs such as ConceptNet (Zhong et al., 2018b) (see column 3 in Table 1).

WikiQA: is an open-domain answer selection dataset (Yang et al., 2015). It was constructed from Bing queries and candidate answer sentences from Wikipedia articles. It contains 1040/140/293 questions in train/development/test, and each question has an average of 9.6 candidate answer sentences.

Results and discussion: Tables 1 and 2 summarize the performance of multiple AHE variants, compared against several baselines and previous works, on two datasets. We draw several observations from these:

(1) The mostly unsupervised AHE, i.e., with the only supervised component being the InferSent embeddings, has solid and stable performance across the three datasets: best on ARC Easy, second best on ARC Challenge (see lines 18 – 21 in Table 1), and top three on WikiQA for MRR (see lines 21 – 24 in Table 2). We find these results encouraging: AHE outperforms many complex supervised neural approaches, including methods having multiple RNNs and stacked attention layers (Wang et al., 2017b; He and Lin, 2016; Miller et al., 2016; Yin et al., 2015; Miao et al., 2016; Musa et al., 2018; Mihaylov et al., 2018), despite the fact that it relies mostly on simple, unsupervised components.

(2) AHE ports well between different partitions (Easy and Challenge) of same dataset (ARC), unlike many of the previous approaches. For example, neural architectures that perform well on ARC Challenge perform worse than a simple IR baseline on ARC Easy (see, e.g., rows 14 and 15 in Table 1) or vice versa (see lines 9 – 12). This lack of portability occurs despite these models being trained/tested within the same partition in Table 1. To emphasize this issue, we explore more aggressive domain transfer settings in Section 5.2.

(3) Ablation analysis – The alignment performance from individual components of AHE are shown

in the baseline blocks of Tables 1 and 2, while the combinations of AHE's components are shown in the corresponding unsupervised and supervised blocks, i.e.: rows 5–8 in table 1 and rows 4–7 in table 2 show performance from individual embeddings of AHE, while rows 18–23 and rows 21–25, respectively, show performance from combinations of AHE components. This comparison indicates that the combination of two or more embedding types are always better than individual embeddings. Further, we see that word embeddings such as GloVe are useful for ARC Easy but not for the Challenge partition of ARC (row 19). In contrast, sentence-level embeddings (InferSent) show the opposite behavior (row 20), suggesting that the more complex the task, the more high-level representations are required.

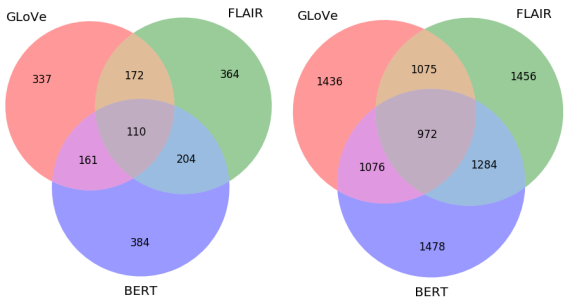
(4) The oracle system (line 24 in table 1 and line 26 in table 2) indicates that the different representations of text are to a large extent complementary: when selecting the correct answer when at least one of the representations proposes it, the oracle system achieves 85.1 P@1 on ARC Easy, 68.1 P@1 on ARC Challenge, and 93.71 MAP on WikiQA. The supervised AHE, which uses a feed-forward neural network to learn when to trust each representation demonstrates that (some of) this complementarity can be learned: the supervised AHE consistently outperforms its unsupervised counterpart, albeit by small amounts. Further, line 23 in Table 1 indicates that additional information about the questions (i.e., grade information) is beneficial, as it provides the meta-classifier more grounding on when to trust which representation. We analyze this complementarity further in Section 5.1.

5 Analysis

To explore the potential of AHE and further understand its individual components, we conducted the following analyses:

5.1 Complementarity of Representations

We calculated the overlap of questions answered correctly by each component of AHE to investigate the complementarity of the different representations. The results are visualized in Figure 4. For simplicity, the figure shows the number of questions answered correctly by the first three (unsupervised) components of AHE, but we found similar trends for the InferSent as well. As shown in the figure, the overlap between any two components is



ARC Challenge ARC Easy

Figure 4: Overlap of correct questions answered by AHE models when they operate over different embeddings. This was a post-hoc analysis on the test partitions; we observed similar trends on training and development. This figure is best viewed in color.

within the range [42 - 53]% in the Challenge partition (GloVe and BERT overlap = (161/384 = 42%) , FLAIR and BERT overlap = (204/384 = 53%)) and [73 - 86]% in the Easy partition. Our current meta-classifier only begins to mine this complementarity, but it is limited because it has no information about the question and candidate answers (other than their scores). We conjecture that considerable performance improvements are possible when such a meta-classifier includes additional information such as question type, question encoding, etc. Our initial results that include grade information (line 23 in Table 1) support this hypothesis. We leave a further exploration of this direction as future work.

5.2 Domain Transfer

As shown in Table 1 and discussed in the previous section, many supervised neural methods do not perform robustly across different partitions (Easy and Challenge) of the same ARC dataset, even though they were trained within each partition. This raises the question of how stable is their performance when trained/tested in different domains, which is closer to a real-world deployment scenario? To answer this question, we trained and tested two state-of-the-art neural models, BiLSTM Max-out (Mihaylov et al., 2018; Conneau et al., 2017) and BiMPM (Wang et al., 2017b), across three domains: ARC Easy, ARC Challenge, and WikiQA. We selected these two approaches because of they are end-to-end neural methods, and they achieve good performance on all datasets. Further, BiMPM is reminiscent of a supervised alignment method, since it computes the overall similarity of question and answers by aligning the

Train \Test	ARC Easy (P@1)	ARC Challenge (P@1)	WikiQA (MAP, MRR)
ARC Easy	34.26, 38.84	23.12, 24.10	(38.71, 40.51), (52.13, 53.87)
ARC Challenge	27.02, 36.17	33.87, 26.39	(39.05, 40.68), (40.09,41.48)
WikiQA	25.84, 38.40	24.32, 25.36	(67.40, 69.08), (69.20, 71.19)
Unsupervised AHE	64.60	33.87	(67.31, 68.53)

Table 3: Performance of two neural QA methods, BiLSTM Max-out and BiMPM, when trained/tested across datasets. The first value in each cell corresponds to BiLSTM Max-out, and the second to BiMPM. The last row contains the best unsupervised performance of AHE, which was *not* trained on any of these three datasets.

tokens' LSTM hidden states.

The results are summarized in Table 3. The table highlights that the performance of these systems varies considerably based on the training domain, even underperforming a random baseline in some configurations. In contrast, the unsupervised AHE does not require training, and obtains state-of-the-art, stable performance across the three datasets. This analysis suggests that future QA evaluations should consider domain transfer as another evaluation measure, to quantify the performance of QA systems under realistic scenarios.

5.3 Brief Qualitative Analysis

We manually analyzed the questions answered incorrectly by AHE and observed that many of the candidate answers were partially answering the questions. As shown in Figure 5, candidate answers 2 and 5 are partially answering the question, while candidate answers 1 and 3 provide topically relevant information. To select the correct answer in such complex questions, especially for short questions, a successful method would have to incorporate inference, e.g., recognizing process questions such as the one in the figure and coupling with it with a dedicated problem solving method (Clark et al., 2013). We leave the integration of inference methods with AHE as future work.

Question - how a water pump works?

1. *A large, electrically driven pump (electropump) for waterworks near the Hengsteysee , Germany.*
2. *A pump is a device that moves fluids (liquids or gases), or sometimes slurries , by mechanical action.*
3. *Pumps can be classified into three major groups according to the method they use to move the fluid: direct lift, displacement, and gravity pumps.*
4. **Pumps operate by some mechanism (typically reciprocating or rotary), and consume energy to perform mechanical work by moving the fluid.**
5. *Pumps operate via many energy sources, including manual operation, electricity, engines , or wind power .*

Figure 5: A question with correct answer in bold font from the WikiQA dataset, which was incorrectly answered by AHE, BiLSTM Max-out and BiMPM.

6 Conclusion

We proposed a simple, mostly-unsupervised alignment model for non-factoid QA, which operates over multiple contextualized embedding representations that model the text at different levels of abstraction. Despite its simplicity, our approach obtains good performance (top three or higher) that is stable across three QA datasets. Our analysis indicates that the different levels of abstraction (character, word, sentence) capture distinct semantics. We showed that this can be modeled with a meta-classifier that learns when and how much to trust the predictions over each representation, and that this has a beneficial impact on performance.

All in all, our work indicates that the first, and possibly best, investment in the design of a QA system should be on contextualized embeddings rather than custom, complex neural architectures.

When such embeddings are available, state-of-the-art performance that is competitive with modern neural approaches for QA can be obtained with simple alignment-based aggregation strategies. Minimally, our work should be regarded as a new, strong baseline for non-factoid question answering or answer sentence selection.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers program, grant number W911NF1810014. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Steven Bird. 2006. Nltk: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL 2006, pages 69–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Rishav Chakravarti, Jiri Navratil, and Cicero Nogueira dos Santos. 2017. Improved answer selection with pre-trained word embeddings. *arXiv preprint arXiv:1708.04326*.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Peter Clark, Philip Harrison, and Niranjan Balasubramanian. 2013. A study of the knowledge base requirements for passing an elementary science test. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 37–42. ACM.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 16–23. Association for Computational Linguistics.

David Golub and Xiaodong He. 2016. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tables as semi-structured knowledge for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 474–483.

Tomasz Jurczyk, Michael Zhai, and Jinho D Choi. 2016. Selqa: A new benchmark for selection-based question answering. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 820–827. IEEE.

Tom Kenter and Maarten De Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM.

Sun Kim, Nicolas Fiorini, W John Wilbur, and Zhiyong Lu. 2017. Bridging the gap: Incorporating a semantic similarity measure for effectively mapping pubmed queries to documents. *Journal of biomedical informatics*, 75:122–127.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Weikang Li, Wei Li, and Yunfang Wu. 2018. A unified model for document-based question answering based on human-like reading strategy. In *AAAI*.

Christopher D Manning. 2015. Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Ryan Musa, Xiaoyan Wang, Achille Fokoue, Nicholas Mattei, Maria Chang, Pavan Kapanipathi, Bassem Makni, Kartik Talamadupula, and Michael Witbrock. 2018. Answering science exam questions using query rewriting with background knowledge. *arXiv preprint arXiv:1809.05726*.

Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. 2018. Learning to attend on essential terms: An enhanced retriever-reader model for scientific question answering. *CoRR*, abs/1808.09492.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. *arXiv preprint arXiv:1704.04572*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *Computing Research Repository, abs/1602.03609*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2).

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2017. Ranking kernels for structures and embeddings: A hybrid preference and classification model. In *EMNLP*.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017a. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *arXiv preprint arXiv:1507.02628*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016a. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016b. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Vikas Yadav, Rebecca Sharp, and Mihai Surdeanu. 2018. Sanity check: A strong alignment and information retrieval baseline for question answering.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Yuyu Zhang, Hanjun Dai, Kamil Toraman, and Le Song. 2018. Kg$\hat{2}$: Learning to reason science exam questions with contextual knowledge graph embeddings. *CoRR*, abs/1805.12393.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2018a. Improving question answering by commonsense-based pre-training. *arXiv preprint arXiv:1809.03568*.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2018b. Improving question answering by commonsense-based pre-training. *CoRR*, abs/1809.03568.