Creating Causal Embeddings for Question Answering with Minimal Supervision

Anonymous EMNLP submission

Abstract

A common model for question answering (QA) is that a good answer is one that is closely related to the question, where relatedness is often determined using general-purpose lexical models such as word embeddings. We argue that a better approach is to look for answers that are related to the question in *the right way*, which may be determined through task-specific embeddings. With causality as use case, we implement this insight in three steps. First, we generate causal embeddings cost-effectively by bootstrapping cause-effect pairs extracted from free text using a small set of seed patterns. Second, we train dedicated embeddings over this data, by using task-specific contexts, i.e., the context of a cause is its effect. Finally, we extend a state-of-the-art reranking approach for QA to incorporate these causal embeddings. We evaluate the causal embedding models both *directly* with a word similarity task, and *indirectly*, in a downstream causal QA task using data from Yahoo! Answers. We show that explicitly modeling causality improves performance in both tasks. In the QA task our best model achieves 37.3% P@1, significantly outperforming a strong baseline by 7.7% (relative).

1 Introduction

Question answering (QA), i.e., finding short answers to natural language questions, is one of the most important but challenging tasks on the road towards natural language understanding (Etzioni, 2011). A common approach for QA is to prefer answers that are closely related to the question, where relatedness is often determined using lexical semantic models such as word embeddings (Yih et al., 2013; Jansen et al., 2014; Fried et al., 2015). While appealing for its robustness to natural language variation, this one-size-fits-all approach does not take into account the wide range of distinct question types that can appear in any given question set, and that are best addressed individually (Chu-Carroll et al., 2004; Ferrucci et al., 2010; Clark et al., 2013).

Given the variety of question types, we suggest that a better approach is to look for answers that are related to the question *through the appropriate relation*, e.g., a causal question should have a cause-effect relation with its answer. If we adopt this view, and continue to work with embeddings as a mechanism for assessing relationship, this raises a key question: how do we train and use task-specific embeddings cost-effectively? Using causality as a use case, we answer this question with a framework for producing causal word embeddings with minimal supervision, and a demonstration that such task-specific embeddings significantly benefit causal QA.

In particular, the contributions of this work are:

(1) A methodology for generating causal embeddings cost-effectively by bootstrapping cause-effect pairs extracted from free text using a small set of seed patterns, e.g., *X causes Y*. We then train dedicated embedding (as well as two other distributional similarity) models over this data. Levy and Goldberg (2014) have modified the algorithm of Mikolov et al. (2013) to use an arbitrary, rather than linear, context. Here we make this context task-specific, i.e., the context of a cause is its effect. Further, to mitigate sparsity and noise, our models are bidirectional, and noise aware (by incorporating the likelihood of noise in the training process).

(2) The insight that QA benefits from task-specific embeddings. We implement a QA system that uses the above causal embeddings to answer questions and demonstrate that they significantly improve performance over a strong baseline. Further, we show that causal embeddings encode complementary information to vanilla embeddings, even when trained from the same knowledge resources.

(3) An analysis of direct vs. indirect evaluations for task-specific word embeddings. We evaluate our causal models both *directly*, in terms of measuring their capacity to rank causally-related word pairs

over word pairs of other relations, as well as *indirectly* in the downstream causal QA task. In both tasks, our analysis indicates that including causal models significantly improves performance. However, from the direct evaluation, it is difficult to estimate which models will perform best in real-world tasks. Our analysis re-enforces recent observations about the limitations of word similarity evaluations (Faruqui et al., 2016): we show that they have limited coverage and may align poorly with real-world tasks.

2 Related Work

Addressing the need for specialized solving methods in QA, Oh et. al (2013) incorporate a dedicated causal component into their system, and note that it improves the overall performance. However, their model is limited by the need for lexical overlap between a causal construction found in their knowledge base and the question itself. Here, we develop a causal QA component that exploits specialized word embeddings to gain robustness to lexical variation.

There has been a vast body of work which demonstrates that word embeddings derived from distributional similarity are useful in many tasks, including question answering – see *inter alia* (Fried et al., 2015; Yih et al., 2013). However, Levy and Goldberg (2015) note that there are limitations on the type of semantic knowledge which is encoded in these general-purpose similarity embeddings. Therefore, here we build customized task-specific embeddings for causal QA.

Customized embeddings have been created for a variety of tasks, including semantic role labeling (FitzGerald et al., 2015; Woodsend and Lapata, 2015), and binary relation extraction (Riedel et al., 2013). Similar to Riedel et al., we train embeddings customized for specific relations, but we bootstrap training data using minimal supervision (i.e., a small set of patterns) rather than relying on distant supervision and large existing knowledge bases. Additionally, while Riedel et al. represent all relations in a general embedding space, here we train a dedicated embedding space for just the causal relations.

In QA, embeddings have been customized to have question words that are close to either their answer words (Bordes et al., 2014), or to structured knowledge base entries (Yang et al., 2014). While these methods are useful for QA, they do not distinguish between different types of questions, and as such their embeddings are not specific to a given question type.

Additionally, embeddings have been customized to distinguish functional similarity from relatedness (Levy and Goldberg, 2014; Kiela et al., 2015). In particular, Levy and Goldberg train their embeddings by replacing the standard linear context of the target word with context derived from the syntactic dependency graph of the sentence. In this work, we make use of this extension to arbitrary context in order to train our embeddings with contexts derived from binary causal relations. We extract cause-effect text pairs such that the cause text becomes the *target* text and the effect text serves as the *context*.

Recently, Faruqui et al.(2016) discussed issues surrounding the evaluation of similarity word embeddings, including the lack of correlation between their performance on word-similarity tasks and "downstream" or real-world tasks like QA, text classification, etc. As they advocate, in addition to a direct evaluation of our causal embeddings, we also evaluate them independently in a downstream QA task. We provide the same comparison for two alternative approaches (an alignment model and a convolutional neural network model), confirming that the direct evaluation performance can be misleading without the task-specific, downstream evaluation.

With respect to extracting causal relations from text, Girju et al. (2002) use modified Hearst patterns (Hearst, 1992) to extract a large number of potential cause-effect tuples, where both causes and effects must be nouns. However, Cole et al. (2005) show that these nominal-based causal relations account for a relatively small percentage of all causal relations, and for this reason, (Yang and Mao, 2014) allow for more elaborate argument structures in their causal extraction by identifying verbs, and then following the syntactic subtree of the verbal arguments to construct their candidate causes and effects. Additionally, Do et al. (2011) observe that nouns as well as verbs can signal causality. We follow these intuitions in developing our causal patterns by using both nouns and verbs to signal potential participants in causal relations, and then allowing for the entire

dominated structures to serve as the cause and/or effect arguments.

3 Approach

Our focus is on reranking answers to causal questions using using task-specific distributional similarity methods. Our approach operates in three steps:

(1) We start by bootstrapping a large number of cause-effect pairs from free text using a small number of syntactic and surface patterns (Section 4).

(2) We then use these bootstrapped pairs to build several task-specific embedding (and other distributional similarity) models (Section 5). We evaluate these models directly on a causal-relation identification task (Section 6).

(3) Finally, we incorporate these models into a reranking framework for causal QA and demonstrate that the resulting approach performs better than the reranker without these task-specific models, even if trained on the same data (Section 7).

4 Extracting Cause-Effect Tuples

Because the success of embedding models depends on large training datasets (Sharp et al., 2015), and such datasets do not exist for open-domain causality, we opted to bootstrap a large number of cause-effect pairs from a small set of patterns. We wrote these patterns using Odin (Valenzuela-Escárcega et al., 2016), a rule-based information extraction framework which has the distinct advantage of being able to operate over multiple representations of content (i.e., surface and syntax). For this work, we make use of rules that operate over both surface sequences as well as dependency syntax in the grammars introduced in steps (2) and (3) below.

Odin operates as a cascade, allowing us to implement a two-stage approach. First, we identify potential participants in causal relations, i.e., the potential causes and effects, which we term **causal mentions (CM)**. A second grammar then identifies actual causal relations that take these CMs as arguments.

We consider both noun phrases (NP) as well as entire clauses to be potential CMs, since causal patterns form around participants that are syntactically more complex than flat NPs. For example, in the sentence *The collapse of the housing bubble caused stock prices to fall*, both the cause (*the collapse of the housing bubble*) and effect (*stock prices to fall*) are more complicated nested structures. Reducing these arguments to non-recursive NPs (e.g., *The collapse* and *stock prices*) is clearly insufficient to capture the relevant context.

Formally, we extract our causal relations using the following algorithm:

(1) Pre-processing: Much of the text we use to extract causal relation tuples comes from the Annotated Gigaword (Napoles et al., 2012). This text is already fully annotated and no further processing is necessary. We additionally use text from the Simple English Wikipedia[1], which we processed using the Stanford CoreNLP toolkit (Manning et al., 2014) and the dependency parser of Chen and Manning (2014).

(2) CM identification: We extract causal mentions (which are able to serve as arguments in our causal patterns) using a set of rules designed to be robust to the variety that exists in natural language. Namely, to find CMs that are noun phrases, we first find words that are tagged as nouns, then follow outgoing dependency links for modifiers and attached prepositional phrases[2], to a maximum depth of two links. To find CMs that are clauses, we first find words that are tagged as verbs (excluding verbs which themselves were considered to signal causation[3]), then again follow outgoing dependency links for modifiers and arguments. We used a total of four rules to label CMs.[4]

(3) Causal tuple extraction: After CMs are identified, a grammar scans the text for causal relations that have CMs as arguments. Different patterns have varying probabilities of signaling causation (Khoo et al, 1998). To minimize the noise in the extracted pairs, we restrict ourselves to a set of 13 rules designed to find unambiguously causal patterns, such

---

[1] https://simple.wikipedia.org/wiki/Main_Page

[2] The outgoing dependency links from the nouns which we followed were: `nn`, `amod`, `advmod`, `ccmod`, `dobj`, `prep_of`, `prep_with`, `prep_for`, `prep_into`, `prep_on`, `prep_to`, and `prep_in`.

[3] The verbs we excluded were: *cause, result, lead, create*.

[4] The two complete grammars, for both CM and causal relation identification, were submitted as supplemental material.

Corpus	Extracted Tuples
Annotated Gigaword	798,808
Simple English Wikipedia	16,425
Total	815,233

Table 1: Number of causal tuples extracted from each corpus.

as *CAUSE led to EFFECT*, where *CAUSE* and *EF-FECT* are CMs. The rules operate by looking for a *trigger* phrase, e.g., *led*, and then following the dependency paths to and/or from the trigger phrase to see if all required CM arguments exist.

Applying this causal grammar over Gigaword and Simple English Wikipedia produced 815,233 causal tuples, as summarized in Table 1. As bootstrapping methods are typically noisy, we manually evaluated the quality of approximately 250 of these pairs selected at random. Of the tuples evaluated, approximately 44% contained some amount of noise. For example, from the sentence *Except for Springer's show, which still relies heavily on confrontational topics that lead to fistfights virtually every day...*, while ideally we would only extract (*confrontational topics → fistfights*), instead we extract the tuple (*show which still relies heavily on confrontational topics → fistfights virtually every day*), which contains a large amount of noise: *show, relies, heavily*, etc. This finding prompted our noise-aware model described in Section 5, which follows.

5 Models

We use the extracted causal tuples to train three distinct distributional similarity models that explicitly capture causality.

Causal Embedding Model (cEmbed): The first distributional similarity model we use is based on the skip-gram word-embedding algorithm of Mikolov et al. (2013), which has been shown to improve a variety of language processing tasks including QA (Yih et al., 2013; Fried et al., 2015). In particular, we use the variant implemented by Levy and Goldberg (2014) which modifies the original algorithm to use an arbitrary, rather than linear, context. Our novel contribution is to make this context task-specific: intuitively, the context of a cause is its effect. Further, these contexts are generated from tuples that are themselves bootstrapped, which minimizes the amount of supervision necessary.

The Levy and Goldberg model trains using single-
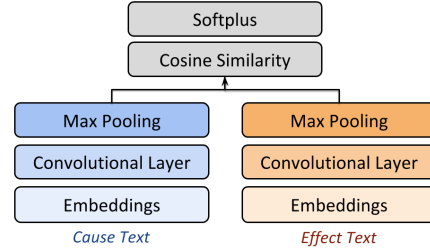
Figure 1: Architecture of the causal convolutional network.

word pairs, while our CMs could be composed of multiple words. For this reason, we decompose each cause–effect tuple, (CM_c, CM_e), such that each word $w_c \in CM_c$ is paired with each word $w_e \in CM_e$. After filtering the extracted cause-effect tuples for stop words and retaining only nouns and verbs, we generated over 3.6M (w_c, w_e) word-pairs[5] from the approximately 800K causal tuples.

The model learns two embedding vectors for each word, one for when the word serves as a target word and another for when the word serves as a context word. Here, since the relation of interest is inherently directional, both sets of embeddings are meaningful, and so we make use of both – the target vectors encode the effects of given causes, whereas the context vectors capture the causes of the corresponding effects.

Causal Alignment Model (cAlign): Monolingual alignment (or translation) models have been shown to be successful in QA (Berger et al., 2000; Echihabi and Marcu, 2003; Soricut and Brill, 2006; Riezler et al., 2007; Surdeanu et al., 2011; Yao et al., 2013), and recent work has shown that they can be successfully trained with less data than embedding models (Sharp et al., 2015).

To verify these observations in our context, we train an alignment model that "translates" causes (i.e., the "source language") into effects (i.e., the "destination language"), using our cause–effect tuples. This is done using IBM Model 1 (Brown et al., 1993) and GIZA++ (Och and Ney, 2003).

Causal Convolutional Neural Network Model (cCNN): Each of the previous models have at their root a bag-of-words representation, which is a simplification of the causality task. To address this potential limitation, we additionally trained a convo-

---

[5]For all models proposed in this section we used lemmas rather than words.

4

lutional neural network (CNN) which operates over variable-length texts, and maintains distinct embeddings for causes and effects. The architecture of this approach is shown in Figure 1, and consists of two sub-networks (one for cause text and one for effect text), each of which begins by converting the corresponding text into 50-dimensional embeddings. These are then fed to a convolutional layer,[6] which is followed by a max-pooling layer of equal length. Then, these top sub-network layers, which can be thought of as a type of phrasal embedding, are merged by taking their cosine similarity. Finally, this cosine similarity is normalized by feeding it into a dense layer with a single node which has a softplus activation. We train the network using a binary cross entropy objective function and the Adam optimizer (Kingma and Ba, 2014), using the Keras library (Chollet, 2015) operating over Theano (Theano Development Team, 2016), a popular deep-learning framework.[7]

Noise-aware Causal Embedding Model (cEmbedNoise): We designed a variant of our cEmbed approach to address the potential impact of the noise introduced by our bootstrapping method. While training, we weigh the causal tuples by the likelihood that they are truly causal. For this, we first score the tuples by their causal pointwise mutual information (PMI) and then scale these scores by the overall frequency of the tuple (Riloff, 1996), to account for the PMI bias toward low-frequency items. That is, the score S of a tuple, t, is computed as:

$$S(t) = \log \frac{p(t|causal)}{p(t)} * \log(freq(t)) \qquad (1)$$

We then discretize these scores into five quantiles, ascribing a linearly decreasing weight during training to items in lower scoring quantiles.

---

[6]The convolutional layer contained 100 filters, had a filter length of 2 (i.e., capturing bigram information), and an inner ReLU activation.

[7]We also experimented with an equivalent architecture where the sub-networks are implemented using long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), and found that they consistently under-perform this CNN architecture. Our conjecture is that CNNs perform better because LSTMs are more sensitive to overall word order than CNNs, which capture only local contexts, and we have relatively little training data, which prevents the LSTMs from generalizing well.

6 Direct Evaluation: Ranking Word Pairs

We begin the assessment of our models with a *direct* evaluation to determine whether or not the proposed approaches capture causality better than general-purpose word embeddings and whether their robustness improves upon a simple database look-up. For this evaluation, we follow the protocol of Levy and Goldberg (2014). In particular, we create a collection of word pairs, half of which are causally related, with the other half consisting of other relations. These pairs are then ranked by our models and several baselines, with the goal of ranking the causal pairs above the others. The embedding models rank the pairs using the cosine similarity between the target vector for the causal word and the context vector of the effect word. The alignment model ranks pairs using the probability $P(\text{Effect}|\text{Cause})$ given by IBM Model 1, and the CNN ranks pairs by the value of the output returned by the network.

6.1 Data

In order to avoid bias towards our extraction methods, we evaluate our models on an external set of word pairs drawn from the SemEval 2010 Task 8 (Hendrickx et al., 2009), a multi-way classification of semantic relations between nominals. We used a total of 1730 nominal pairs, 865 of which were from the Cause-Effect relation (e.g., (*dancing* → *happiness*)) and an equal number which were randomly selected from the other eight relations (e.g., (*juice* → *grapefruit*), from the Entity-Origin relation). This set was then randomly divided into equally-sized development and test partitions.

6.2 Baselines

We compared our distributional similarity models against three baselines:

Vanilla Embeddings Model (vEmbed): a standard `word2vec` model trained with the skip-gram algorithm and a sliding window of 5, using the same texts from which our causal pairs were extracted.[8] As with the cEmbed model, SemEval pairs were ranked using the cosine similarity between the vector representations of their arguments.

---

[8]All embedding models analyzed here, including this baseline and our causal variants, produced embedding vectors of 200 dimensions.

5
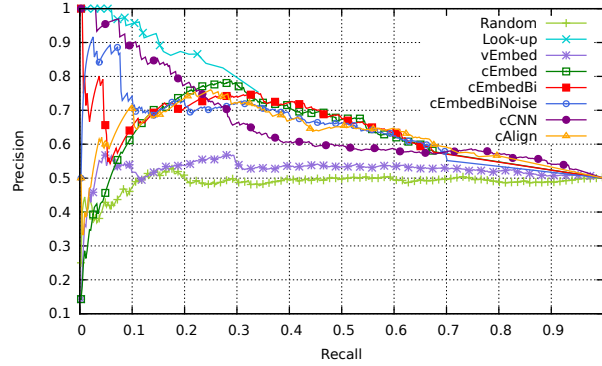
Figure 2: Precision-recall curve showing the ability of each model to rank causal pairs above non-causal pairs. For clarity, we do not plot cEmbedNoise, which performs worse than cEmbedBiNoise. The Look-up model has no data points beyond the 35% recall point.

Look-up Baseline: a given SemEval pair was ranked by the number of times it appeared in our extracted cause-effect tuples.

Random: pairs were randomly shuffled.

6.3 Results

Figure 2 shows the precision-recall (PR) curve for each of the models and baselines. As expected, the causal models are better able to rank causal pairs than the vanilla embedding baseline (vEmbed), which, in turn, outperforms the random baseline. Our look-up baseline, which ranks pairs by their frequency in our causal database, shows a high precision for this task, but has coverage for only 35% of the causal SemEval pairs.

Some models perform better on the low-recall portion of the curve (e.g., the look-up baseline and cCNN), while the embedding and alignment models have a higher and more consistent performance across the PR curve. We hypothesize that models that better *balance* precision and recall will perform better in a real-world QA task, which may need to access a given causal relation through a variety of lexical patterns or variations. We empirically validate this observation in Section 7.

The PR curve for the causal embeddings shows an atypical dip at low-recall. To examine this, we analyzed the top-ranked 15% of SemEval pairs, and found that incorrectly ranked pairs were not found in the database of causal tuples. Instead, these incorrect rankings were largely driven by low frequency words whose embeddings could not be robustly estimated due to lack of direct evidence. Because this sparsity is partially driven by directionality, we im-

plemented a bidirectional embedding model (cEmbedBi) that (a) trains a second embedding model by reversing the input (effects as targets, causes as contexts), and (b) ranks pairs by the *average* of the scores returned by these two unidirectional causal embedding models. As Figure 2 shows, the bidirectional embedding variants consistently outperform their unidirectional counterparts. All in all, the best overall model is cEmbedBiNoise, which is both bidirectional and incorporates the noise handling approach from Section 5. This model substantially improves performance in the low-recall portion of the curve, while also showing strong performance across the curve.

7 Indirect Evaluation: QA Task

The main objective of our work is to investigate the impact of a customized causal embedding model for QA. Following our direct evaluation, which solely evaluated the degree to which our models directly encode causality, here we evaluate each of our proposed causal models in terms of their contribution to a downstream real-world QA task.

Our QA system uses a standard reranking approach (Jansen et al., 2014). In this architecture, the candidate answers are initially extracted and ranked using a shallow candidate retrieval (CR) component that uses solely information retrieval techniques, then they are re-ranked using a "learning to rank" approach. In particular, we used SVM rank[9], a Support Vector Machine adapted for ranking, and re-ranked the candidate answers with a set features de-

---

[9] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

6

rived from both the initial CR score and the models we have introduced.

7.1 Data

We evaluate on a set of causal questions extracted from the Yahoo! Answers corpus[10] with simple surface patterns such as *What causes ...* and *What is the result of ...*[11]. We extracted a total of 3031 questions, each with at least four candidate answers, and we evaluated performance using five-fold cross-validation, with three folds for training, one for development, and one for testing.

7.2 Models and Features

We evaluate the contribution of the bidirectional and noise-aware causal embedding models (cEmbedBi, and cEmbedBiNoise) as well as the causal alignment model (cAlign) and the causal CNN (cCNN). These models are compared against three baselines: the vanilla embeddings (vEmbed), the lookup baseline (LU), and additionally a vanilla alignment model (vAlign) which is trained over 65k question-answer pairs from Yahoo! Answers.

The features[12] we use for the various models are:

Embedding model features: For both our vanilla and causal embedding models, we use the same set of features as Fried et al. (2015): the maximum, minimum, and average pairwise cosine similarity between question and answer words, as well as the overall similarity between the composite question and answer vectors. When using the causal embeddings, since the relation is directed, we first determine whether the question text is the cause or the effect[13], which in turn determines which embeddings to use for the question text and which to use for the candidate answer texts. For example, in a question such as "*What causes X?*", since *X* is the effect, all cosine similarities would be found using the effect

vectors for the question words and the cause vectors for the answer candidate words.

Alignment model features: We use the same global alignment probability, $p(Q|A)$ of Surdeanu et al. (2011). In our causal alignment model, we adapt it to causality by using $p(\text{Effect}|\text{Cause})$. Again, for the causal model, we first determine the direction of the causal relation implied in the question. We include the additional undirected alignment features based on Jensen-Shannon distance, proposed more recently by Fried et al. (2015), in our vanilla alignment model. However, due to the directionality inherent in causality, they do not apply to our causal model so there we omit them.

Look-up feature: For the look-up baseline we count the number of times words from the question and answer appear together in our database of extracted causal pairs, once again after determining the directionality of the questions. If the total number of matches is over a threshold[14], we consider the causal relation to be established and give the candidate answer a score of 1; or a score of 0, otherwise.

7.3 Results

The overall results are summarized in Table 2. Lines 1–5 in the table show that each of our baselines performed better than CR by itself, except for vAlign, suggesting that the vanilla alignment model does not generate accurate predictions for causal questions. The strongest baseline was CR + vEmbed (line 3), the vanilla embeddings trained over Gigaword, at 34.6% P@1. For this reason, we consider this to be the baseline to "beat", and perform statistical significance of all proposed models with respect to it.

Individually, the cEmbedBi model is the best performing of the causal models. While the performance of cAlign in the direct evaluation was comparable to that of cEmbedBi, here it performs far worse (line 6 vs 8), suggesting that the robustness of embeddings is helpful in QA. Notably, despite the strong performance of the cCNN in the low-recall portion of the PR curve in the direct evaluation, here the model performs poorly (line 9).

---

[10]Freely available through Yahoo!'s Webscope program (`research-data-requests@yahoo-inc.com`)

[11]We lightly filtered these with stop words to remove non-causal questions, such as those based on math problems and the results of sporting events. Our dataset will be freely available, conditioned on users having obtained the Webscope license.

[12]Due to the variety of features used, each feature described here is independently normalized to lie between 0.0 and 1.0.

[13]We do this through the use of simple regular expressions, e.g., "^ [Ww]hat ([a-z]+){0,3}cause.+"

[14]Empirically determined to be 100 matches. Note that using this threshold performed better than simply using the total number of matches.

#	Model	P@1
	Baselines	
1	Random	16.43
2	CR	24.31
3	CR + vEmbed	34.61
4	CR + vAlign	19.24
5	CR + Look-up (LU)	29.56
	Single Causal Models	
6	CR + cEmbedBi	31.32
7	CR + cEmbedBiNoise	30.15
8	CR + cAlign	23.49
9	CR + cCNN	24.66
	Model Combinations	
10	CR + vEmbed + cEmbedBi	37.08*
11	CR + vEmbed + cEmbedBiNoise	35.50*
12	CR + vEmbed + cEmbedBi + LU	36.75*
13	CR + vEmbed + cAlign	34.31
14	CR + vEmbed + cCNN	33.45
	Model Stacking	
15	CR + vEmbed + cEmbedBi + cEmbedBiNoise	37.28*

Table 2: Performance in the QA evaluation, measured by precision-at-one (P@1). The "Bi" suffix indicates a bidirectional model; the "Noise" suffix indicates a model that is noise aware. * indicates that the difference between the corresponding model and the CR + vEmbed baseline is statistically significant ($p < 0.05$), determined through a one-tailed bootstrap resampling test with 10,000 iterations.

No individual causal model outperforms the strong vanilla embedding baseline (line 4), likely owing to the reduction in generality inherent to building task-specific QA models. However, comparing lines 6–9 vs. 10–14 shows that the vanilla and causal models are capturing different and complementary kinds of knowledge (i.e., causality vs. association through distributional similarity), and are able to be combined to increase overall task performance (lines 10–12). These results highlight that QA is a complex task, where solving methods need to address the many distinct information needs in question sets, including both causal and direct association relations. This contrasts with the direct evaluation, which focuses strictly on causality, and where the vanilla embedding baseline performs near chance. This observation highlights one weakness of word similarity tasks: their narrow focus may not directly translate to estimating their utility in real-world NLP applications.

Adding in the lookup baseline (LU) to the best-performing causal model does not improve performance (compare lines 10 and 12), suggesting that the bidirectional causal embeddings subsume the contribution of the LU model. cEmbedBi (line 10) also performs better than cEmbedBiNoise (line 11). We conjecture that the "noise" filtered out by cEmbedBiNoise contains distributional similarity information, which is useful for the QA task. cEmbedBi vastly outperforms cCNN (line 14), suggesting that strong overall performance across the precision-recall curve better translates to the QA task. We hypothesize that the low cCNN performance is caused by insufficient training data, preventing the CNN architecture from generalizing well.

Our best performing overall model combines both variants of the causal embedding model (cEmbedBi and cEmbedBiNoise), reaching a P@1 of 37.3%, which shows a 7.7% relative improvement over the strong CR + vEmbed baseline.

8 Conclusion

Here we present a framework for creating customized embeddings tailored to the information need of causal questions. We train three popular models (embedding, alignment, and CNN) using causal tuples extracted with minimal supervision by bootstrapping cause-effect pairs from free text, and evaluate their performance both directly (i.e., the degree to which they capture causality), and indirectly (i.e., their real-world utility on a high-level question answering task).

We show that models that incorporate a knowledge of causality perform best for both tasks. Our analysis suggests that the models that perform best in the real-world QA task are those that have consistent performance across the precision-recall curve in the direct evaluation. In QA, where the vocabulary is much larger, precision must be balanced with high-recall, and this is best achieved by our causal embedding model. Additionally, we show that vanilla and causal embedding models address different information needs of questions, and can be combined to improve performance.

Extending this work beyond causality, we hypothesize that additional embedding spaces customized to the different information needs of questions would allow for robust performance over a larger variety of questions, and that these customized embedding models should be evaluated both directly and indirectly to accurately characterize their performance.

8

References

[Berger et al.2000] A. Berger, R. Caruana, D. Cohn, D. Freytag, and V. Mittal. 2000. Bridging the lexical chasm: Statistical approaches to answer finding. In *Proc. of the 23rd Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*.

[Bordes et al.2014] A. Bordes, S. Chopra, and J. Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

[Brown et al.1993] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

[Chen and Manning2014] D. Chen and C. D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proc. of the Conferenc on Empirical Methods for Natural Language Processing (EMNLP)*.

[Chollet2015] F. Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

[Chu-Carroll et al.2004] J. Chu-Carroll, K. Czuba, J. M. Prager, A. Ittycheriah, and S. Blair-Goldensohn. 2004. IBM's PIQUANT II in TREC 2004. In *Text Retrieval Conference (TREC)*.

[Clark et al.2013] P. Clark, P. Harrison, and N. Balasubramanian. 2013. A study of the knowledge base requirements for passing an elementary science test. In *Proc. of the 2013 workshop on Automated Knowledge Base Construction (AKBC)*, pages 37–42.

[Cole et al.2005] S. V. Cole, M. D. Royal, M. G. Valtorta, M. N. Huhns, and J. B. Bowles. 2005. A lightweight tool for automatically extracting causal relationships from text. In *SoutheastCon, 2006. Proc. of the IEEE*, pages 125–129.

[Do et al.2011] Q. X. Do, Y. S. Chan, and D. Roth. 2011. Minimally supervised event causality identification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 294–303.

[Echihabi and Marcu2003] A. Echihabi and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16–23.

[Etzioni2011] O. Etzioni. 2011. Search needs a shake-up. *Nature*, 476(7358):25–26.

[Faruqui et al.2016] M. Faruqui, Y. Tsvetkov, R. Rastogi, and C. Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

[Ferrucci et al.2010] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. 2010. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79.

[FitzGerald et al.2015] N. FitzGerald, O. Täckström, K. Ganchev, and D. Das. 2015. Semantic role labeling with neural network factors. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 960–970.

[Fried et al.2015] D. Fried, P. Jansen, G. Hahn-Powell, M. Surdeanu, and P. Clark. 2015. Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, 3:197–210.

[Girju and Moldovan2002] R. Girju and D. I. Moldovan. 2002. Text mining for causal relations. In *FLAIRS Conference*, pages 360–364.

[Hearst1992] M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th conference on Computational linguistics (COLING)*, pages 539–545.

[Hendrickx et al.2009] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proc. of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99.

[Hochreiter and Schmidhuber1997] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Jansen et al.2014] P. Jansen, M. Surdeanu, and P. Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Khoo et al.1998] C. S.G. Khoo, J. Kornfilt, R. N. Oddy, and S. H. Myaeng. 1998. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Literary and Linguistic Computing*, 13(4):177–186.

[Kiela et al.2015] D. Kiela, F. Hill, and S. Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Kingma and Ba2014] D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Levy and Goldberg2014] O. Levy and Y. Goldberg. 2014. Dependency-based word embeddings. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 302–308.

[Levy et al.2015] O. Levy, S. Remus, C. Biemann, I. Dagan, and I. Ramat-Gan. 2015. Do supervised distributional methods really learn lexical inference relations. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[Manning et al.2014] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Mikolov et al.2013] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Napoles et al.2012] C. Napoles, M. Gormley, and B. Van Durme. 2012. Annotated gigaword. In *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100.

[Och and Ney2003] F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Oh et al.2013] J.-H. Oh, K. Torisawa, C. Hashimoto, M. Sano, S. De Saeger, and K. Ohtake. 2013. Why-question answering using intra-and inter-sentential causal relations. In *The 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1733–1743.

[Riedel et al.2013] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proc. of the Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

[Riezler et al.2007] S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 464–471.

[Riloff1996] E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 1044–1049.

[Sharp et al.2015] R. Sharp, P. Jansen, M. Surdeanu, and P. Clark. 2015. Spinning straw into gold. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*.

[Soricut and Brill2006] R. Soricut and E. Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Journal of Information Retrieval - Special Issue on Web Information Retrieval*, 9(2):191–206.

[Surdeanu et al.2011] M. Surdeanu, M. Ciaramita, and H. Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383.

[Theano Development Team2016] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

[Valenzuela-Escárcega et al.2016] M. A. Valenzuela-Escárcega, G. Hahn-Powell, and M. Surdeanu. 2016. Odin's runes: A rule language for information extraction. In *Proc. of the 10th International Conference on Language Resources and Evaluation (LREC)*.

[Woodsend and Lapata2015] K. Woodsend and M. Lapata. 2015. Distributed representations for unsupervised semantic role labeling. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Yang and Mao2014] X. Yang and K. Mao. 2014. Multi level causal relation identification using extended features. *Expert Systems with Applications*, 41(16):7171–7181.

[Yang et al.2014] M.-C. Yang, N. Duan, M. Zhou, and H.-C. Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650.

[Yao et al.2013] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark. 2013. Semi-markov phrase-based monolingual alignment. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Yih et al.2013] W. Yih, M. Chang, C. Meek, and A. Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*.