Represent, Aggregate, and Constrain: A Novel Architecture for Machine Reading from Noisy Sources

Abstract

In order to extract event information from text, a machine reading model must learn to accurately read and interpret the ways in which that information is expressed. But it must also, as the human reader must, aggregate numerous individual value hypotheses into a single coherent global analysis, applying global constraints which reflect prior knowledge of the domain. In this work we focus on the task of extracting plane crash event information from clusters of related news articles whose labels are derived via distant supervision. Unlike previous machine reading work, we assume that while most target values will occur frequently in most clusters, they may also be missing or incorrect. We introduce a novel neural architecture to explicitly model the noisy nature of the data and to deal with these aforementioned learning issues. Our models are trained end-to-end and achieve an improvement of more than 12.1 F$_1$ over previous work, despite using far less linguistic annotation. We apply factor graph constraints to promote more coherent event analyses, with belief propagation inference formulated within the transitions of a recurrent neural network. We show this technique additionally improves maximum F$_1$ by up to 2.8 points, resulting in a relative improvement of 50% over the previous state-of-the-art.

Introduction

Recent work in the area of machine reading has focused on learning in a scenario with perfect information. Whether identifying target entities for simple cloze style queries BIBREF0 , BIBREF1 , or reasoning over short passages of artificially generated text BIBREF2 , short stories BIBREF3 , or children's stories BIBREF4 , these systems all assume that the corresponding text is the unique source of information necessary for answering the query – one that not only contains the answer, but does not contain misleading or otherwise contradictory information.

For more practical question answering, where an information retrieval (IR) component must first fetch the set of relevant passages, the text sources will be less reliable and this assumption must be discarded. Text sources may vary in terms of their integrity (whether or not they are intentionally misleading or unreliable), their accuracy (as in the case of news events, where a truthful but outdated article may contain incorrect information), or their relevance to the query. These characteristics necessitate not only the creation of high-precision readers, but also the development of effective strategies for aggregating conflicting stories into a single cohesive account of the event.

Additionally, while many question answering systems are designed to extract a single answer to a single query, a user may wish to understand many aspects of a single entity or event. In machine reading, this is akin to pairing each text passage with multiple queries. Modeling each query as an independent prediction can lead to analyses that are incoherent, motivating the need to model the dependencies between queries.

We study these problems through the development of a novel machine reading architecture, which we apply to the task of event extraction from news cluster data. We propose a modular architecture which decomposes the task into three fundamental sub-problems: (1) representation INLINEFORM0 scoring, (2) aggregation, and (3) global constraints. Each corresponds to an exchangeable component of our model. We explore a number of choices for these components, with our best configuration improving performance by INLINEFORM1 F INLINEFORM2 , a INLINEFORM3 relative improvement, over the previous state-of-the-art.

The Case for Aggregation

Effective aggregation techniques can be crucial for identifying accurate information from noisy sources. Figure FIGREF1 depicts an example of our problem scenario. An IR component fetches several

documents based on the query, and sample sentences are shown for each document. The goal is to extract the correct value, of which there may be many mentions in the text, for each slot. Sentences in INLINEFORM0 express a target slot, the number of fatalities, but the mention corresponds to an incorrect value. This is a common mistake in early news reports. Documents INLINEFORM1 and INLINEFORM2 also express this slot, and with mentions of the correct value, but with less certainty.

A model which focuses on a single high-scoring mention, at the expense of breadth, will make an incorrect prediction. In comparison, a model which learns to correctly accumulate evidence for each value across multiple mentions over the entire cluster can identify the correct information, circumventing this problem. Figure FIGREF1 (bottom) shows how this pooling of evidence can produce the correct cluster-level prediction.

Model

In this section we describe the three modeling components of our proposed architecture:

We begin by defining terminology. A news cluster INLINEFORM0 is a set of documents, INLINEFORM1 , where each document is described by a sequence of words, INLINEFORM2 . A mention is an occurrence of a value in its textual context. For each value INLINEFORM3 , there are potentially many mentions of INLINEFORM4 in the cluster, defined as INLINEFORM5 . These have been annotated in the data using Stanford CoreNLP BIBREF5 .

Representations and Scoring

For each mention INLINEFORM0 we construct a representation INLINEFORM1 of the mention in its context. This representation functions as a general “reading” or encoding of the mention, irrespective of

the type of slots for which it will later be considered. This differs from some previous machine reading research where the model provides a query-specific reading of the document, or reads the document multiple times when answering a single query BIBREF0 . As in previous work, an embedding of a mention's context serves as its representation. We construct an embedding matrix INLINEFORM2 , using pre-trained word embeddings, where INLINEFORM3 is the dimensionality of the embeddings and INLINEFORM4 the number of words in the cluster. These are held fixed during training. All mentions are masked and receive the same one-hot vector in place of a pretrained embedding. From this matrix we embed the context using a two-layer convolutional neural network (CNN), with a detailed discussion of the architecture parameters provided in Section SECREF4 . CNNs have been used in a similar manner for a number of information extraction and classification tasks BIBREF6 , BIBREF7 and are capable of producing rich sentence representations BIBREF8 .

Having produced a representation INLINEFORM0 for each mention INLINEFORM1 , a slot-specific attention mechanism produces INLINEFORM2 , representing the compatibility of mention INLINEFORM3 with slot INLINEFORM4 . Let INLINEFORM5 be the representation matrix composed of all INLINEFORM6 , and INLINEFORM7 is the index of INLINEFORM8 into INLINEFORM9 . We create a separate embedding, INLINEFORM10 , for each slot INLINEFORM11 , and utilize it to attend over all mentions in the cluster as follows: DISPLAYFORM0

The mention-level scores reflect an interpretation of the value's encoding with respect to the slot. The softmax normalizes the scores over each slot, supplying the attention, and creating competition between mentions. This encourages the model to attend over mentions with the most characteristic contexts for each slot.

Aggregating Mention-level Scores

For values mentioned repeatedly throughout the news cluster, mention scores must be aggregated to produce a single value-level score. In this section we describe (1) how the right aggregation method can better reflect how the gold labels are applied to the data, (2) how domain knowledge can be incorporated into aggregation, and (3) how aggregation can be used as a dynamic approach to identifying missing information.

In the traditional view of distant supervision BIBREF9 , if a mention is found in an external knowledge base it is assumed that the mention is an expression of its role in the knowledge base, and it receives the corresponding label. This assumption does not always hold, and the resulting spurious labels are frequently cited as a source of training noise BIBREF10 , BIBREF11 . However, an aggregation over all mention scores provides a more accurate reflection of how distant supervision labels are applied to the data.

If one were to assign a label to each mention and construct a loss using the mention-level scores (INLINEFORM0) directly, it would recreate the hard labeling of the traditional distant supervision training scenario. Instead, we relax the distant supervision assumption by using a loss on the value-level scores (INLINEFORM1), with aggregation to pool beliefs from one to the other. This explicitly models the way in which cluster-wide labels are applied to mentions, and allows for spuriously labeled mentions to receive lower scores, “explaining away” the cluster's label by assigning a higher score to a mention with a more suitable representation.

Two natural choices for this aggregation are max and sum. Formally, under max aggregation the value-level scores for a value INLINEFORM0 and slot INLINEFORM1 are computed as: DISPLAYFORM0

And under sum aggregation: DISPLAYFORM0

If the most clearly expressed mentions correspond to correct values, max aggregation can be an effective strategy BIBREF10 . If the data set is noisy with numerous spurious mentions, a sum aggregation favoring values which are expressed both clearly and frequently may be the more appropriate choice.

The aforementioned aggregation methods combine mention-level scores uniformly, but for many domains one may have prior knowledge regarding which mentions should more heavily contribute to the aggregate score. It is straightforward to augment the proposed aggregation methods with a separate weight INLINEFORM0 for each mention INLINEFORM1 to create, for instance, a weighted sum aggregation: DISPLAYFORM0

These weights may be learned from data, or they may be heuristically defined based on a priori beliefs. Here we present two such heuristic methods.

News articles naturally deviate from the topical event, often including comparisons to related events, and summaries of past incidents. Any such instance introduces additional noise into the system, as the contexts of topical and nontopical mention are often similar. Weighted aggregation provides a natural foothold for incorporating topicality into the model.

We assign aggregation weights heuristically with respect to a simple model of discourse. We assume every document begins on topic, and remains so until a sentence mentions a nontopical flight number. This and all successive sentences are considered nontopical, until a sentence reintroduces the topical flight. Mentions in topical sentences receive aggregation weights of INLINEFORM0 , and those in non-topical sentences receive weights of INLINEFORM1 , removing them from consideration completely.

In the aftermath of a momentous event, news outlets scramble to release articles, often at the expense of providing accurate information.

We hypothesize that the earliest articles in each cluster are the most likely to contain misinformation, which we explore via a measure of information content. We define the information content of an article as the number of correct values which it mentions. Using this measure, we fit a skewed Gaussian distribution over the ordered news articles, assigning INLINEFORM0 , where INLINEFORM1 is the smoothed information content of INLINEFORM2 as drawn from the Gaussian.

A difficult machine reading problem unique to noisy text sources, where the correct values may not be present in the cluster, is determining whether to predict any value at all. A common solution for handling such missing values is the use of a threshold, below which the model returns null. However, even a separate threshold for each slot would not fully capture the nature of the problem.

Determining whether a value is missing is a trade-off between two factors: (1) how strongly the mention-level scores support a non-null answer, and (2) how much general information regarding that event and that slot is given. We incorporate both factors by extending the definition of INLINEFORM0 and its use in Eq. EQREF9 –Eq. to include not only mentions, but all words. Each non-mention word is treated as a mention of the null value: DISPLAYFORM0

where INLINEFORM0 is the set of mentions. The resulting null score varies according to both the cluster size and its content. Smaller clusters with fewer documents require less evidence to predict a non-null value, while larger clusters must accumulate more evidence for a particular candidate or a null value will be proposed instead.

The exact words contained in the cluster also have an effect. For example, clusters with numerous mentions of killed, died, dead, will have a higher INLINEFORM0 Fatalities INLINEFORM1 , requiring the model to be more confident in its answer for that slot during training. Additionally, this provides a mechanism for driving down INLINEFORM2 when INLINEFORM3 is not strongly associated with

INLINEFORM4 .

Global Constraints

While the combination of learned representations and aggregation produces an effective system in its own right, its predictions may reflect a lack of world knowledge. For instance, we may want to discourage the model from predicting the same value for multiple slots, as this is not a common occurrence.

Following recent work in computer vision which proposes a differentiable interpretation of belief propagation inference BIBREF12 , BIBREF13 , we present a recurrent neural network (RNN) which implements inference under this constraint.

A factor graph is a graphical model which factorizes the model function using a bipartite graph, consisting of variables and factors. Variables maintain beliefs over their values, and factors specify scores over configurations of these values for the variables they neighbor.

We constrain model output by applying a factor graph model to the INLINEFORM0 scores it produces. The slot INLINEFORM1 taking the value INLINEFORM2 is represented in the factor graph by a Boolean variable INLINEFORM3 . Each INLINEFORM4 is connected to a local factor INLINEFORM5 whose initial potential is derived from INLINEFORM6 . A combinatorial logic factor, Exactly-1 BIBREF14 , is (1) created for each slot, connected across all values, and (2) created for each value, connected across all slots. This is illustrated in Figure FIGREF22 . Each Exactly-1 factor provides a hard constraint over neighboring Boolean variables requiring exactly one variable's value to be true, therefore diminishing the possibility of duplicate predictions during inference.

The resulting graph contains cycles, preventing the use of exact message passing inference. We instead

treat an RNN as implementing loopy belief propagation (LBP), an iterative approximate message passing inference algorithm. The hidden state of the RNN is the set of variable beliefs, and each round of message updates corresponds to one iteration of LBP, or one recurrence in the RNN.

There are two types of messages: messages from variables to factors and messages from factors to variables. The message that a variable INLINEFORM0 sends to a factor INLINEFORM1 (denoted INLINEFORM2) is defined recursively w.r.t. to incoming messages from its neighbors INLINEFORM3 as follows: DISPLAYFORM0

and conveys the information “My other neighbors jointly suggest I have the posterior distribution INLINEFORM0 over my values.” In our RNN formulation of message passing the initial outgoing message for a variable INLINEFORM1 to its neighboring Exactly-1 factors is: DISPLAYFORM0

where the sigmoid moves the scores into probability space.

A message from an Exactly-1 factor to its neighboring variables is calculated as:

INLINEFORM0

All subsequent LBP iterations compute variable messages as in Eq. EQREF24 , incorporating the out-going factor beliefs of the previous iteration.

Data

The Stanford Plane Crash Dataset BIBREF15 is a small data set consisting of 80 plane crash events, each paired with a set of related news articles. Of these events, 40 are reserved for training, and 40 for

testing, with the average cluster containing more than 2,000 mentions. Gold labels for each cluster are derived from Wikipedia infoboxes and cover up to 15 slots, of which 8 are used in evaluation (Figure TABREF54).

We follow the same entity normalization procedure as reschke2014, limit the cluster size to the first 200 documents, and further reduce the number of duplicate documents to prevent biases in aggregation. We partition out every fifth document from the training set to be used as development data, primarily for use in an early stopping criterion. We also construct additional clusters from the remaining training documents, and use this to increase the size of the development set.

Experiments

In all experiments we train using adaptive online gradient updates (Adam, see kingma2014). Model architecture and parameter values were tuned on the development set, and are as follows (chosen values in bold):

The number of training epochs is determined via early stopping with respect to the model performance on development data. The pre-trained word embeddings are 200-dimensional GLoVe embeddings BIBREF16 .

Systems

We evaluate on four categories of architecture:

reschke2014 proposed several methods for event extraction in this scenario. We compare against three notable examples drawn from this work:

[leftmargin=*]

Reschke CRF: a conditional random field model.

Reschke Noisy-OR: a sequence tagger with a "Noisy-OR" form of aggregation that discourages the model from predicting the same value for multiple slots.

Reschke Best: a sequence tagger using a cost-sensitive classifier, optimized with SEARN BIBREF17 , a learning-to-search framework.

Each of these models uses features drawn from dependency trees, local context (unigram/part-of-speech features for up to 5 neighboring words), sentence context (bag-of-word/part-of-speech), words/part-of-speech of words occurring within the value, as well as the entity type of the mention itself.

The representation and scoring components of our architecture, with an additional slot for predicting a null value. The INLINEFORM0 scores are used when constructing the loss and during decoding. These scores can also be aggregated in a max/sum manner after decoding, but such aggregation is not incorporated during training.

Representation, scoring, and aggregation components, trained end-to-end with a cluster-level loss. Null

values are predicted as described in Sec. UID18 .

kadlec2016 present AS Reader, a state-of-the-art model for cloze-style QA. Like our architecture, AS Reader aggregates mention-level scores, pooling evidence for each answer candidate. However, in cloze-style QA an entity is often mentioned in complementary contexts throughout the text, but are frequently in similar contexts in news cluster extraction.

We tailor AS Reader to event extraction to illustrate the importance of choosing an aggregation which reflects how the gold labels are applied to the data. EE-AS Reader is implemented by applying Eq. EQREF9 and Eq. to each document, as opposed to clusters, as documents are analogous to sentences in the cloze-style QA task. We then concatenate the resulting vectors, and apply sum aggregation as before.

Evaluation

We evaluate configurations of our proposed architecture across three measures. The first is a modified version of standard precision, recall, and F INLINEFORM0 , as proposed by reschke2014. It deviates from the standard protocol by (1) awarding full recall for any slot when a single predicted value is contained in the gold slot, (2) only penalizing slots for which there are findable gold values in the text, and (3) limiting candidate values to the set of entities proposed by the Stanford NER system and included in the data set release. Eight of the fifteen slots are used in evaluation. Similarly, the second evaluation measure we present is standard precision, recall, and F INLINEFORM1 , specifically for null values.

We also evaluate the models using mean reciprocal rank (MRR). When calculating the F INLINEFORM0

-based evaluation measure we decode the model by taking the single highest-scoring value for each slot. However, this does not necessarily reflect the quality of the overall value ranking produced by the model. For this reason we include MRR, defined as: DISPLAYFORM0

where rank INLINEFORM0 is the rank position of the first correct value for a given cluster and slot pair INLINEFORM1 , and INLINEFORM2 , the number of such pairs, is INLINEFORM3 , the product of the total number of clusters with the total number of predicted slots.

Results

Results are presented in Table TABREF44 . In comparison to previous work, our any configuration of our RAC architecture with sum-based aggregation outperforms the best existing systems by a minimum of 9.8 F INLINEFORM0 . In comparison to the various Mention-CNN systems, it is clear that this improvement is not a result of different features or the use of pre-trained word embeddings, or even the representational power of the CNN-based embeddings. Rather, it is attributable to training end-to-end with aggregation and a cluster-level loss function.

With respect to aggregation, sum-based methods consistently outperform their max counterparts, indicating that exploiting the redundancy of information in news clusters is beneficial to the task. The topic-based aggregation is statistically significant improvement over standard sum aggregation (p INLINEFORM0), and produces the highest performing unconstrained system.

Date-based aggregation did not yield a statistically significant improvement over sum aggregation. We hypothesize that the method is sound, but accurate datelines could only be extracted for 31 INLINEFORM0 documents. We did not modify the aggregation weights (INLINEFORM1) for the remaining documents, minimizing the effect of this approach.

The EE-AS Reader has the lowest overall performance, which one can attribute to pooling evidence in a manner that is poorly suited to this problem domain. By placing a softmax over each document's beliefs, what is an advantage in the cloze-style QA setting here becomes a liability: the model is forced to predict a value for every slot, for every each document, even when few are truly mentioned.

Effects of Global Constraints

In Table TABREF50 we show the results of incorporating factor graph constraints into our best-performing system. Performing one iteration of LBP inference produces our highest performance, an F INLINEFORM0 of 44.9. This is 14.9 points higher than Reschke's best system, and a statistically significant improvement over the unconstrained model (p INLINEFORM1). The improvements persist throughout training, as shown in Figure FIGREF52 .

Additional iterations reduce performance. This effect is largely due to the constraint assumption not holding absolutely in the data. For instance, multiple slots can have the null value, and zero is common value for a number of slots. Running the constraint inference for a single iteration encourages a 1-to-1 mapping from values to slots, but it does not prohibit it. This result also implies that a hard heuristic decoding constraint time would not be as effective.

Error Analysis

We randomly selected 15 development set instances which our best model predicts incorrectly. Of these, we find three were incorrectly labeled in the gold data as errors from the distance supervision hypothesis (i.e., “zero chance” being labeled for 0 survivors, when the number of survivors was not mentioned in the cluster), and should not be predicted. Six were clearly expressed and should be predictable, with highly correlated keywords present in the context window, but were assigned low scores by the model. We

belief a richer representation which combines the generalization of CNNs with the discrete signal of n-gram features BIBREF18 may solve some of these issues.

Four of the remaining errors appear to be due to aggregation errors. Specifically, the occurrence of a particular punctuation mark with far greater than average frequency resulted in it being predicted for three slots. While these could be filtered out, a more general solution may be to build a representation based on the actual mention (“Ryanair”), in addition to its context. This may reduce the scores of these mentions to such an extent that they are removed from consideration.

Table TABREF54 shows the accuracy of the model on each slot type. The model is struggles with predicting the Injuries and Survivors slots. The nature of news media leads these slots to be discussed less frequently, with their mentions often embedded more deeply in the document, or expressed textually. For instance, it is common to express INLINEFORM0 =Survivors, INLINEFORM1 as “no survivors”, but it is impossible to predict a 0 value in this case, under the current problem definition.

Connections to Pointer Networks

A pointer network uses a softmax to normalize a vector the size of the input, to create an output distribution over the dictionary of inputs BIBREF23 . This assumes that the input vector is the size of the dictionary, and that each occurrence is scored independently of others. If an element appears repeatedly throughout the input, each occurrence is in competition not only with other elements, but also with its duplicates.

Here the scoring and aggregation steps of our proposed architecture can together be viewed as a pointer network where there is a redundancy in the input which respects an underlying grouping. Here the softmax normalizes the scores over the input vector, and the aggregation step again yields an output

distribution over the dictionary of the input.

Conclusion and Future Work

In this work we present a machine reading architecture designed to effectively read collections of documents in noisy, less controlled scenarios where information may be missing or inaccurate. Through attention-based mention scoring, cluster-wide aggregation of these scores, and global constraints to discourage unlikely solutions, we improve upon the state-of-the-art on this task by 14.9 F INLINEFORM0 .

In future work, the groundwork laid here may be applied to larger data sets, and may help motivate the development of such data. Larger noisy data sets would enable the differentiable constraints and weighted aggregation to be included during the optimization, and tuned with respect to data. In addition, we find the incorporation of graphical model inference into neural architectures to be a powerful new tool, and potentially an important step towards incorporating higher-level reasoning and prior knowledge into neural models of NLP.