Textual Analogy Parsing: What’s Shared and What’s Compared among Analogous Facts

Matthew Lamm1,3 Arun Tejasvi Chaganty2,3+ Christopher D. Manning1,2,3 Dan Jurafsky1,2,3 Percy Liang2,3
1Stanford Linguistics 2Stanford Computer Science 3Stanford NLP Group
{mlamm,jurafsky}@stanford.edu
{chaganty,manning,pliang}@cs.stanford.edu

Abstract

To understand a sentence like “whereas only 10% of White Americans live at or below the poverty line, 28% of African Americans do” it is important not only to identify individual facts, e.g., poverty rates of distinct demographic groups, but also the higher-order relations between them, e.g., the disparity between them. In this paper, we propose the task of Textual Analogy Parsing (TAP) to model this higher-order meaning. The output of TAP is a frame-style meaning representation which explicitly specifies what is shared (e.g., poverty rates) and what is compared (e.g., White Americans vs. African Americans, 10% vs. 28%) between its component facts. Such a meaning representation can enable new applications that rely on discourse understanding such as automated chart generation from quantitative text. We present a new dataset for TAP, baselines, and a model that successfully uses an ILP to enforce the structural constraints of the problem.

1 Introduction

The task of information extraction by and large seeks to populate a knowledge base with individuated facts extracted from text (Sarawagi, 2008). For example, given the sentence:

(E1) [According to the U.S. Census, whereas only 10% of White Americans live at or below the poverty line today, 28% of African Americans do.]1

one would extract two independent facts about voter registration, about the two distinct demographic groups. On the other hand, the theory of discourse maintains that part of the above sentence’s meaning inheres in the fact that clauses C1

*Author contributed significantly.
1Data in E1 and the figure sentence from Morris (2014).
2 A Semantic Representation of Analogy

Let us revisit the example sentence from the previous section (E1), where a pair of analogous quantitative facts about poverty rates of different demographic groups are presented in contrast. Individually, these can be represented using the semantic role structures in Figure 3, but representing them separately in this way fails to capture the fact that they are analogous, i.e., structurally and semantically similar but distinct.

Instead, we can explicitly show points of similarity and difference between them in the two-tiered frame structure in Figure 2, which we call a TAP frame. The outer tier of the TAP frame contains *shared content*, or information pertinent to all of the facts in question, and the inner tier contains *compared content*, the information that varies across the set of facts.

Mapping from an utterance to a TAP frame requires three types of relational reasoning. Firstly, one must decompose the utterance into a set of facts, where a fact is represented as a set of semantic roles. Then, one must identify the shared content across facts by aligning roles that are semantically equivalent, in the sense that they are either the same span, are coreferent, or are synonymous. For example, in Figure 2 the phrase ‘U.S. Census’ occurs as the source of both facts because it scopes over the entire sentence in which they appear. Additionally, one must identify the compared content by aligning roles that are analogous, in the sense that they are semantically similar but nevertheless distinct. For example, the phrases ‘White Americans’ and ‘African Americans’ are analogous in our running sentence, playing the same role in their respective facts, while signifying distinct demographic groups.
Table 1: Representative sentences from the Quantitative TAP dataset. Co-indexing (e.g., A1/Q1) indicates when spans are part of the same QSRL fact. Parentheses indicate shared content spans and brackets indicate compared content spans. To parse (a), one must recognize that ‘to acquire PS of New Hampshire’ is elided but nevertheless an implied TH(eme) in two of the clauses, and that ‘offered’ and ‘bid’ are contextually synonymous Q(uantities). Moreover, one must note that the A(gents) are analogous, and hence part of the compared content. In (b), ‘First Boston’, ‘UAL’ and ‘worth’, contribute a S(ource), TH(eme), and Q(antity) to the shared content respectively. Here, C(ause) roles are compared content.

<table>
<thead>
<tr>
<th>Train (n = 1000)</th>
<th>Test (n = 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>av. max tot.</td>
</tr>
<tr>
<td></td>
<td>1.4 3 1383</td>
</tr>
<tr>
<td>Length</td>
<td>av. max tot.</td>
</tr>
<tr>
<td></td>
<td>2.6 16 –</td>
</tr>
</tbody>
</table>

Table 2: Dataset statistics (average per sentence, max per sentence, and total over the dataset) for the number of analogy frames (Count) and the number of values compared within each frame (Length).

3 The Quantitative TAP Dataset

Motivated by the application of automated graphical plot generation from text, we annotated a dataset of quantitative TAP frames from the Penn Treebank WSJ corpus (Marcus et al., 1999).

As our SRL representation of quantitative facts, we employ the Quantitative Semantic Role Labeling (QSRL) framework we previously defined in Lamm et al. (2018). Having identified a numerical VALUE in text (e.g., 10%), QSRL asks, “what does this number measure?” to determine its associated QUANTITY (e.g., a poverty rate). It might also identify, for example, the WHOLE out of which this percentage is measured (e.g., the set of African Americans), and the TIME at which the quantity took on the value (e.g., today), etc. We employ all fifteen QSRL roles in our annotations.

Our annotations not only capture the relation between a quantitative predicate and its arguments, but also the higher-order analogy relations between them. The distinction is reflected in the sentences in Table 1 from the dataset: Colored spans are co-indexed when they participate in the same quantitative fact; spans with like roles surrounded by parentheses are shared content, meaning that they are either synonymous or co-referent; spans with like roles surrounded by brackets are compared content, meaning that they are analogous but semantically distinct.

To identify instances of quantitative analogy in the WSJ corpus, we first prune out any sentence having fewer than three numerical mentions, where a numerical mention is defined as a contiguous sequence of CD POS tags. Of those left, we manually identify those containing one or more quantitative analogies, i.e., ones in which numerical values are compared content. We estimate the incidence of these to be around 20%. A linguist then annotated 1,100 of these for analogy relationships. See Table 2 for a summary.

Using an independent set of expert annotations on 100 of these sentences, we measured a significant per-token label agreement of 0.882 and edge label agreement of 0.991 using Krippendorf’s $\alpha$.

Table 1 highlights some of the challenging linguistic phenomena in the data. With respect to identifying the shared content of a TAP frame, these can be coarsely divided into two sets. Firstly, in scope, ellipsis, and gapping, a single syntactic element serves as a role in multiple QSRL frames. This is exemplified by the phrase ‘PS of New Hampshire’ in Table 1(a): It is mentioned explicitly as a THEME of the first fact, and only implied in the second two. Based on a random sample of 100 train sentences, we estimate that 86% of frames in the data exhibit these phenomena. Secondly, in synonymy and coreference, multiple elements appear in a sentence but contribute the same role to the shared content, e.g., ‘offered’ and ‘bid’ in Table 1(a). We estimate that 31% of frames in

---

2High edge agreement should be expected because edges are type-constrained and thus easy to identify. Additionally, we computed agreement after matching overlapping spans.
the data exhibit these phenomena.

One must learn to identify analogy relationships over a diverse set of compared content roles, with distinct semantic properties: in Table 1(a), AGENT is a compared content role, whereas in Table 1(b), CAUSE is.

4 Modeling TAP in the Quantitative Setting

We model TAP by generating a typed analogy graph over spans of an input text that is isomorphic to the set of TAP frames in that text, e.g., Figure 2. Each vertex in the graph corresponds to a role-labeled span, and edges represent semantic relations between them.

In this graph, each fact is uniquely identified by a VALUE vertex, which is connected via a FACT edge to all of its associated roles. Any two shared content vertices across facts are connected by an EQUIVALENCE edge, indicating that they are coreferent or synonymous. A single vertex can also be shared across facts by linking via a FACT edge to more than one VALUE vertex, suggesting a scopal relationship. Finally, any two vertices which are compared content in the graph are linked via an ANALOGY edge.

More formally, given an utterance \( x \) with tokens \( x_1, \ldots, x_n \), let \( G \) be a graph with vertices \( V \) and edges \( E \). For a vertex \( v = (i, j, l) \in V \), \( 1 \leq i < j \leq n \) are the start and end token indices of a span in \( x \) with role \( l \in L_Q \defeq \{ \text{value,}, \ldots, \text{quant} \} \), the set of QSRL roles. For an edge \( e = (v, v', l) \in E \), \( v, v' \in V \) and \( l \in L_R \defeq \{ \text{fact, equivalence, analogy} \} \).

For \( G \) so defined to encode a set of valid TAP frames, it must satisfy certain constraints:

1. Well-formedness constraints. For any two vertices \( v, v' \in V \), their associated spans must not overlap. Furthermore, every vertex must participate in at least one FACT edge, i.e., no disconnected vertices.

2. Typing constraints. FACT relations are always drawn from a VALUE vertex to a non-VALUE vertex. ANALOGY and EQUIVALENCE are only ever drawn between two vertices of the same role.

3. Unique facts. If a VALUE vertex \( v \) is connected to two distinct vertices \( v' \) and \( v'' \) of the same role via a FACT edge, then EQUIVALENCE \((v', v'')\) exists.

4. Transitivity constraints. ANALOGY and EQUIVALENCE edges are transitive: if EQUIVALENCE \((v, v') \in E \) and EQUIVALENCE \((v', v'') \in E \) then EQUIVALENCE \((v, v'') \in E \) also. This also holds for ANALOGY edges, but only when \( v, v' \) and \( v'' \) are VALUE vertices.

5. Analogy. There must be at least one pair of analogous VALUE vertices, and for each such pair, there must be a pair of analogous facts connected to them: if \( v, v' \) are two VALUE vertices with ANALOGY \((v, v') \in E \), then there must also exist \( w, w' \) as two non-VALUE vertices with FACT \((v, w) \in E \), FACT \((v', w') \in E \), ANALOGY \((w, w') \in E \).

Note that while these constraints rely on the choice of VALUE as the role that grounds quantitative facts, they reflect the general idea that analogy is a structured mapping between meaning representations.

5 A Neural and ILP Model for TAP

We now present a neural and ILP model that predicts analogy graphs as defined in Section 4. Given a sentence, the neural model predicts a distribution over role-labeled spans with edges denoting semantic relations between them. Then, we use an ILP to decode while enforcing the TAP constraints defined in Section 4. Figure 4 presents an overview of the architecture.

Context-sensitive word embeddings. We first encode the words in a sentence by embedding each token using fixed word embeddings. We also concatenate a few linguistic features to the word embeddings, such as named entity tags and dependency relations. These features are generated using CoreNLP (Manning et al., 2014) and represented by randomly-initialized, learned embeddings for symbols together with the fixed word embedding of each token’s dependency head and the dependency path length between adjacent tokens. The token embeddings are then passed through several stacked convolutional layers (Kim, 2014). While the first convolutional layer can only capture local information, subsequent layers allow for longer-distance reasoning.

Span prediction. Next, we feed the outputs of a single fully-connected hidden layer to a conditional random field (CRF) (Lafferty et al., 2001),
Figure 4: An overview of the proposed neural model: The sentence embedding represents features across the entire sentence using multiple convolutional layers. We then use a conditional random field (CRF) layer to predict labeled spans $p_m$ and to generate span and edge embeddings. We use a feedforward (FF) layer on the edge embeddings to predict edge labels $p_{mn}$. Together, $p_m$ and $p_{mn}$ form a distribution over edges and labels that we decode into TAP frames.

which defines a joint distribution over per-token role labels. We thus obtain spans from this distribution corresponding to vertices of the graph described in Section 4 by merging contiguous role-labels in the maximum likelihood label sequence predicted by the CRF.

**Edge prediction with PATHMAX features.** For edge prediction, we use the spans identified above to construct span and edge embeddings: for every span $(i, j)$ that was predicted, we construct a span vector $s_m = \sum_{k=i}^j \hat{x}_k$. We also construct a role-label score vector for the span, $p_m$ by summing the role-label probability vectors of its constituent tokens. Then, for every vertex pair $(m, n)$, we construct an edge representation $e_{mn}$. The basis of this representation is simply the concatenation of the span representations, the sum of the span representations, their respective role-label score vectors $p_m$ and $p_{mn}$, and relative token distances.

To capture long-distance phenomena like scope, we also incorporate features into $e_{mn}$ from the dependency paths between the two spans by max-pooling the (learned) dependency relation embeddings along the path between the tokens.\(^3\) When computing the representation between two spans, we take the average of the path embedding between each pair of tokens within them. We call this extension PATHMAX.

The resulting edge representation $e_{mn}$ is passed through a single fully-connected hidden layer and an output layer to predict a distribution over edge labels $p_{mn}$, for each pair of spans.

**Training.** The supervised data described in Section 3 provides gold spans and edges between them. Thus we define a loss function with two terms: one for the log-likelihood of the span labels output by the CRF model, and one for the cross-entropy loss on the edge labels. We train the span and edge components of the model jointly.

**Decoding.** We consider two methods for decoding the span-level and edge-level label distributions $p_m$ and $p_{mn}$ into a labeled graph respecting the constraints described in Section 4.

As a simple greedy method to enforce these constraints, we begin by picking the most likely role for each span and edge and then discarding any edges and spans that violate the well-formedness (1) and typing constraints (2). We then enforce transitivity constraints (4) by incrementally building a cluster of analogous and equivalent spans. We then resolve the unique facts constraint (3) by keeping only the span with highest FACT edge score. Finally, for every cluster of analogous value spans, we check that the analogy constraint (5) holds and if not, discard the cluster.

We also implement an optimal decoder that encodes the TAP constraints as an ILP (Roth and Yih, 2004; Do et al., 2012). The ILP tries to find an optimal decoding according to the model, subject to hard constraints imposed on the solution space. For example, we require that solutions satisfy the ‘connected spans’ constraint:

$$\forall s \exists s' : e(s, s', \text{FACT})$$

In plain English, this says that every span $s$ in a solution must be connected via a FACT edge to some other span $s'$. See the supplementary material for the full list of constraints we employ. We solve the ILPs with Gurobi (Gurobi Optimization, Inc., 2018).

**6 Experiments**

We now describe the experimental setup of our neural model (Section 5) on the dataset of TAP frames we created (Section 3). Results and discussion are reported in Section 7.

\(^3\)The dependency paths are directed but unlexicalized.
Evaluation metrics. The primary metric we use to measure the accuracy of a system on frame prediction is the precision, recall and $F_1$ between the labeled vertex-edge-vertex triples predicted by the model and those in the gold parse. If there are multiple predicted spans that overlap with a single gold span or vice versa, we find a matching of predicted and gold spans that maximizes overlap.

In addition to the primary metric, we also report precision, recall and $F_1$ when predicting labeled (non-VALUE) spans and predicting labeled edges before performing any decoding.\footnote{We exclude VALUE spans from span scores because they are easy to predict and thus inflate model performance.} We also use the matching process described above for both these sets of metrics. Standardized evaluation code is provided with the dataset.

Experimental setup. We compare the neural models presented in Section 5 in addition to a log-linear baseline. The log-linear baseline uses the same fixed word embeddings as the neural model in addition to the named entity and dependency parse features described in Section 5. The key difference is that instead of learning a sentence embedding or hidden layers, the log-linear model simply uses a CRF to predict span labels directly from fixed input features, and then uses a single sigmoid layer to predict edge labels from deterministic edge embeddings, $e_{mn}$.

For the neural models, we used three convolutional layers for sentence embedding with a filter size of 3. Every layer other than the input layer used a hidden dimension of 50 with ReLU nonlinearities. We introduced a single dropout layer ($p = 0.5$) between every two layers in the network (including at the input). We used 50-dimensional GloVe embeddings (Pennington et al., 2014) learned from Wikipedia 2014 and Gigaword 5 as pre-trained word embeddings, and initialized the embeddings for the features randomly. We chose relatively low input- and hidden-vector dimension because of the size of our data. The network was trained for 15 epochs using ADADELTA (Zeiler, 2012) with a learning rate of 1.0. All models were implemented in PyTorch (Paszke et al., 2017).

### 7 Results and Discussion

Frame prediction results on the test set are summarized in Table 3. Our three main findings are that (i) the neural network model far outperforms the log-linear model on our frame metric, (ii) including linguistic features further increases performance, and (iii) so does using an optimal decoder over a greedy method.

Quantitative error analysis. To better understand which aspects of our model contribute to the task, we perform an ablation study on the span and edge predictions of our model prior to decoding. With respect to span prediction (Table 4), we found that the fixed word vectors, along with a CRF, were able to capture the information needed to identify QSRL role-spans. Indeed, the log-linear baseline, which directly uses these word vectors as features for a CRF (i.e., the log-linear model) was sufficient to predict spans.

With respect to edge prediction, we perform an ablation study on the span and edge predictions of our model prior to decoding.

### Table 3: Performance of models on the test data. Combining the neural model with linguistic features and using an optimal decoder to enforce semantic constraints led to the best performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>Feats.</th>
<th>Dec.</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log-linear</td>
<td>✓</td>
<td>gr.</td>
<td>46.3</td>
<td>21.8</td>
<td>29.7</td>
</tr>
<tr>
<td>Log-linear</td>
<td>✓</td>
<td>opt.</td>
<td>37.1</td>
<td>27.5</td>
<td>31.6</td>
</tr>
<tr>
<td>Neural</td>
<td>×</td>
<td>gr.</td>
<td>50.7</td>
<td>38.4</td>
<td>43.7</td>
</tr>
<tr>
<td>Neural</td>
<td>×</td>
<td>opt.</td>
<td>52.8</td>
<td>48.6</td>
<td>50.6</td>
</tr>
<tr>
<td>Neural</td>
<td>✓</td>
<td>gr.</td>
<td>54.9</td>
<td>57.4</td>
<td>56.1</td>
</tr>
<tr>
<td>Neural</td>
<td>✓</td>
<td>opt.</td>
<td>56.4</td>
<td>68.8</td>
<td>62.0</td>
</tr>
</tbody>
</table>

### Table 4: Performance of models on labeled (non-VALUE) span prediction during cross-validation prior to decoding. We found using a CRF to be the most important aspect: simply using fixed word vectors with a CRF (i.e., the log-linear model) was sufficient to predict spans.

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log-linear (all feats.)</td>
<td>42.8</td>
<td>82.3</td>
<td>56.3</td>
</tr>
<tr>
<td>Neural (no feats.)</td>
<td>41.7</td>
<td>79.1</td>
<td>54.6</td>
</tr>
<tr>
<td>Neural (all feats.)</td>
<td>41.5</td>
<td>79.2</td>
<td>54.4</td>
</tr>
<tr>
<td>w/o NER</td>
<td>41.6</td>
<td>79.1</td>
<td>54.5</td>
</tr>
<tr>
<td>w/o dep.</td>
<td>41.2</td>
<td>77.5</td>
<td>53.8</td>
</tr>
<tr>
<td>w/o CRF</td>
<td>36.1</td>
<td>73.1</td>
<td>48.3</td>
</tr>
</tbody>
</table>

In a separate experiment, the neural model outperformed the log-linear model when they were trained only to do span prediction.
Those with a bullish view see the dollar bears, while those with a bearish view see the dollar trading around 1.760 marks. . . While the log-linear model did well at predicting spans, it did a poor job predicting edges, indicating that learning to extract higher-order features from learned span embeddings is necessary for identifying semantic relations between them (Table 5). We also found that linguistic features were important: in particular, we found that syntactic features – the dependency path features (PATHMAX) and dependency labels – played a big role in edge prediction, followed by type information from NER tags.

Qualitative error analysis. Our model is tasked with jointly identifying QSRL parses of analogous facts in a sentence, and ANALOGY and EQUIVALENCE relations among them. As described in Section 4, these pieces interact in mutually constraining ways, and thus it is possible for local errors to have global effects on predicted frames.

In Figure 5, for example, the model correctly identifies the gold TIME spans as part of a TAP frame, but mistakenly predicts that they are linked by EQUIVALENCE, and thus modify the same VALUE span. In the gold parse, they are linked by ANALOGY, and modify distinct VALUE spans. As a result of this misclassification, the model leaves out an entire QSR fact from the resulting parse.

In many cases, the model successfully identifies compared content roles between QSRL facts. In Figure 6, we show an example where it does not manage to do so. Here, unable to identify the ANALOGY relation between the phrases ‘Those with a bullish view’ and ‘the dollar bears’, the model instead chooses two identical sequences ‘the dollar’ as the non-VALUE compared content. Inspecting edge probability scores from the model before decoding reveals that the neural model thinks that the first instance of ‘the dollar’ in the sentence is semantically analogous to the second; it can be confused by surface similarity into classifying ANALOGY relations.

Application to plot generation. As we have seen, textual analogy is frequently used to compare quantities along some axis of differentiation. For example, one might compare the stock prices of different companies, or describe the change in some quantity’s value over time. Such analogy relationships can alternately be expressed in the form of a plot.

Indeed, there is a natural correspondence between charts and TAP frames over quantitative facts: VALUES of a quantitative TAP frame are plotted against other compared content roles, and elements of the shared content correspond with scopal chart elements, such as titles. This mapping is well-defined provided analogous values share.

In Figure 7, we deterministically plot TAP
Investment unit stake in Vickers PLC

Investment unit stake in Vickers PLC

(a) Before constraints
(b) After constraints

Figure 7: Charts generated from TAP frames. Charts (a) and (b) are generated from the sentence ‘Vicker’s PLC . . . raised its stake in the company Friday to 15.02% from about 14.6% Thursday and from 13.6% the previous week.’ Before imposing constraints, the neural model assigns multiple values to the TIME arguments ‘Thursday’ and ‘Friday’, over-extending their scope. Imposing structural constraints ensures the correct assignment of TIMES to VALUES. Charts (c) and (d) are generated from the sentence ‘In the auto sector, Bayerische Motoren Werke plunged 14.5 marks to 529 marks, Daimler-Benz dropped 10.5 to 700, and Volkswagen slumped 9 to 435.5.’ Here, the model fails to associate an absolute (blue) and relative (red) VALUE pair with a THEME role. The imposition of global constraints corrects this, linking them to the THEME ‘Daimler-Benz’.

8 Related Work

Analogy. In the cognitive science literature, analogy is a general form of relational reasoning unique to human cognition (Tversky and Gati, 1978; Holyoak and Thagard, 1996; Goldstone and Son, 2005; Penn et al., 2008; Holyoak, 2012). Our model of textual analogy is particularly influenced by Structure Mapping Theory (Falkenhainer et al., 1989; Gentner and Markman, 1997), an influential cognitive model of analogy as a structure-preserving map between concepts.

Within the NLP community, there has been much work focused on inferring lexical analogies between generic concepts, e.g., tennis:racket::baseball:bat (Mikolov et al., 2013; Turney, 2013), from global distributional statistics. Such analogies are generic, type-level patterns whose structure exists in the nature of the language; here, we are interested in specific analogies whose structure is conveyed by a particular sentence.

Discourse and Information Extraction. TAP is an information extraction task that synthesizes ideas from semantic role labeling on the one hand and discourse parsing on the other. The former produces predicate-argument representations of individual facts in a text (Baker et al., 1998; Gildea and Jurafsky, 2002; Palmer et al., 2005); the latter identifies discourse relations between syntactic clauses (Taboada and Mann, 2006; Prasad et al., 2007; Pitler et al., 2009; Prasad et al., 2010; Surdeanu et al., 2015).

TAP first maps from syntax to a set of SRL-style representations, and then identifies structurally-constrained, higher-order relations among them. It is in this sense reminiscent of, but distinct from, work on causal processes by Berant et al. (2014).

Numbers in NLP. There has been some work on understanding numbers in text. This includes quantitative reasoning (Kushman et al., 2014; Roy et al., 2015), numerical information extraction (Madaan et al., 2016), and techniques for making numbers more easily interpretable in text (Chaganty and Liang, 2016; Kim et al., 2016).

If pursued further, the application of plotting...
quantitative text that we discuss in this paper could help to clarify quantitative text on the web (Larkin and Simon, 1987; Barrio et al., 2016).

**Neural modeling.** Recent work has shown the promise of sophisticated neural models on semantic role labeling (He et al., 2017). Similar to other such sequence prediction models, e.g., those for named entity recognition (Lample et al., 2016) or semantic role labeling (Zhou and Xu, 2015), our span prediction utilizes a neural CRF. Our model also has an edge-prediction component, which benefits from a simplified version of the PathLSTM model of Roth and Lapata (2016). Our edge-prediction model also uses an embedding concatenation component, which was inspired by recent work on neural coreference resolution (Lee et al., 2017). He et al. (2017) also impose semantic constraints during prediction, but use $A^*$ search instead of an ILP.

9 Conclusion

In this paper we have presented a new task, textual analogy parsing, or TAP. Given a sentence about a set of analogous facts, TAP outputs a frame representation that expresses the points of similarity and difference in their meanings.

We note that in the particular case of quantitative text, TAP frames correspond with charts. We develop a new dataset of TAP frames from quantitative newswire, and compare a variety models for TAP. Our best model employs a globally optimal decoder to enforce the structural constraints of analogy; its outputs can be mapped to well-formed charts of quantitative information extracted from text.

We view this work to be an exciting step in the direction of deeper discourse modeling. Future work might further extend the recovery of analogy as part of information extraction. This might include TAP outside of the quantitative domain, or TAP at the paragraph level.

**Acknowledgments**

We would like to thank the members of the Stanford NLP Group for reviewing early versions of the paper, and would also like to thank the anonymous reviewers for their thoughtful feedback.

**References**


Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In EMNLP.


A ILP constraints

The optimal decoder described in Section 5 implements the TAP constraints in Section 4 as an ILP. Here we present a representative set of our ILP constraints in the form of boolean expressions. In the following, variables $s$ refer to spans in an input text. $\mathcal{L}_Q = \{\text{QUANT} \ldots \text{VALUE}\}$ is the set of QSRL roles, and $\mathcal{L}_R = \{\text{EQUIVALENCE}, \text{ANALOGY}, \text{FACT}\}$ is the set of semantic relations between them.

We let $r(s, l) = 1$ if the decoder assigns to $s$ the role-label $l \in \mathcal{L}_Q$, and 0 otherwise, and let $\bar{r}(s) = 1$ if for any role-label $l \in \mathcal{L}_Q \ r(s, l) = 1$, and zero otherwise. Similarly, we let $e(s, s', l) = 1$ if an edge is identified between spans $s, s'$ with label $l \in \mathcal{L}_R$, and let $\bar{e}(s, s') = 1$ if for any edge label $l \in \mathcal{L}_R$, and 0 otherwise.

1. (Unique Roles) $\forall s : \sum_{l \in \mathcal{L}_Q} r(s, l) = \bar{r}(s)$
2. (Unique Edges) $\forall s, s' : \sum_{l \in \mathcal{L}_R} e(s, s', l) = \bar{e}(s, s')$
3. (Connected Spans) $\forall s \exists s' : e(s, s', \text{FACT})$
4. (Active Edges) $\forall s, s' : \bar{e}(s, s') \Rightarrow \bar{r}(s) \land \bar{r}(s')$
5. (Equivalence and Analogy Typing) $\forall l, s, s'$:
   - $e(s, s', \text{EQUIVALENCE}) \Rightarrow \exists l : r(s, l) \land r(s', l)$
   - $e(s, s', \text{ANALOGY}) \Rightarrow \exists l : r(s, l) \land r(s', l)$
6. (Fact Typing) $\forall s, s' : e(s, s', \text{FACT}) \Rightarrow (r(s, \text{VALUE}) \land \neg r(s', \text{VALUE})) \lor$ $(r(s', \text{VALUE}) \land \neg r(s, \text{VALUE}))$
7. (Equality and Analogy Triangles) $\forall s, s', s''$:
   - $e(s, s', \text{EQUIVALENCE}) \land e(s', s'', \text{EQUIVALENCE}) \Rightarrow e(s, s'', \text{EQUIVALENCE})$
   - $e(s, s', \text{ANALOGY}) \land e(s', s'', \text{ANALOGY}) \Rightarrow e(s, s'', \text{ANALOGY})$ but only when $r(s, \text{VALUE}) \land r(s', \text{VALUE}) \land r(s'', \text{VALUE})$
8. (Fact-Equality) $\forall l, s, s', s''$:
   - $e(s, s', \text{FACT}) \land e(s', s'', \text{EQUIVALENCE}) \Rightarrow e(s, s'', \text{FACT})$
9. (Analogy Quadrangle) $\forall s, s' : r(s, \text{VALUE}) \land r(s', \text{VALUE}) \land e(s, s', \text{ANALOGY}) \Rightarrow \exists s'', s''' : e(s, s'', \text{FACT}) \land e(s', s''', \text{FACT}) \land e(s'', s''', \text{ANALOGY})$