How Much is 131 Million Dollars? Putting Numbers in Perspective with Compositional Descriptions

Arun Tejasvi Chaganty
Computer Science Department
Stanford University
chaganty@cs.stanford.edu

Percy Liang
Computer Science Department
Stanford University
pliang@cs.stanford.edu

Abstract

How much is 131 million US dollars? To help readers put such numbers in context, we propose a new task of automatically generating short descriptions known as perspectives, e.g. "$131 million is about the cost to employ everyone in Texas over a lunch period". First, we collect a dataset of numeric mentions in news articles, where each mention is labeled with a set of rated perspectives. We then propose a system to generate these descriptions consisting of two steps: formula construction and description generation. In construction, we compose formulae from numeric facts in a knowledge base and rank the resulting formulas based on familiarity, numeric proximity and semantic compatibility. In generation, we convert a formula into natural language using a sequence-to-sequence recurrent neural network. Our system obtains a 15.2% F_1 improvement over a non-compositional baseline at formula construction and a 12.5 BLEU point improvement over a baseline description generation.

1 Introduction

When posed with a mention of a number, such as "Cristiano Ronaldo, the player who Madrid acquired for [...] a *$131 million*" (Figure 1), it is often difficult to comprehend the scale of large (or small) absolute values like *$131 million* (Paulos, 1988; Seife, 2010). Studies have shown that providing relative comparisons, or *perspectives*, such as "about the cost to employ everyone in Texas over a lunch period" significantly improves comprehension when measured in terms of memory retention or outlier detection (Barrio et al., 2016).

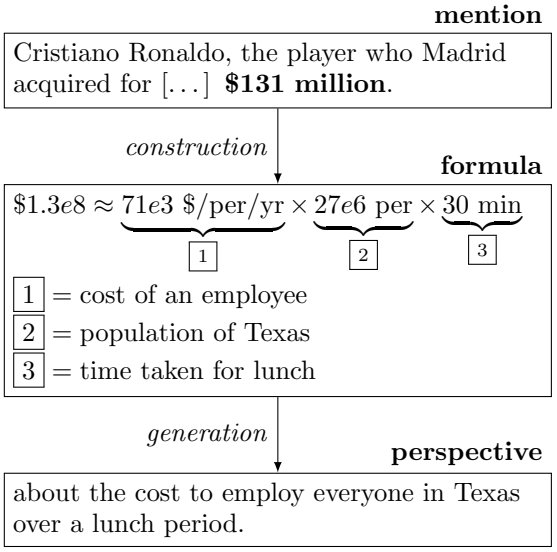

Figure 1: An overview of the perspective generation task: given a *numeric mention*, generate a short description (a *perspective*) that allows the reader to appreciate the scale of the mentioned number. In our system, we first construct a *formula* over facts in our knowledge base and then generate a description of that formula.

Previous work in the HCI community has relied on either manually generated perspectives (Barrio et al., 2016) or present a fact as is from a knowledge base (Chiacchieri, 2013). As a result, these approaches are limited to contexts in which a relevant perspective already exists.

In this paper, we generate perspectives by composing facts from a knowledge base. For example, we might describe *$100,000* to be "about twice the median income for a year", and describe *$5 million* to be the "about how much the average person makes over their lifetime". Leveraging compositionality allows us to achieve broad coverage of numbers from a relatively small collection of familiar facts, e.g. median income and a person's

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 578–587,
Berlin, Germany, August 7-12, 2016.

lifetime.

Using compositionality in perspectives is also concordant with our understanding of how people learn to appreciate scale. Jones and Taylor (2009) find that students learning to appreciate scale do so mainly by *anchoring* with familiar concepts, e.g. $50,000 is slightly less than the median income in the US, and by *unitization*, i.e. improvising a system of units that is more relatable, e.g. using the Earth as a measure of mass when describing the mass of Jupiter to be that of 97 Earths. Here, compositionality naturally unitizes the constituent facts: in the examples above, money was unitized in terms of median income, and time was unitized in a person's lifetime. Unitization and anchoring have also been proposed by Chevalier et al. (2013) as the basis of a design methodology for constructing visual perspectives called concrete scales.

When generating compositional perspectives, we must address two key challenges: constructing familiar, relevant and meaningful formulas and generating easy-to-understand descriptions or perspectives. We tackle the first challenge using an overgenerate-and-rank paradigm, selecting formulas using signals from familiarity, compositionality, numeric proximity and semantic similarity. We treat the second problem of generation as a translation problem and use a sequence-to-sequence recurrent neural network (RNN) to generate perspectives from a formula.

We evaluate individual components of our system quantitatively on a dataset collected using crowdsourcing. Our formula construction method improves on F_1 over a non-compositional baseline by about 17.8%. Our generation method improves over a simple baseline by 12.5 BLEU points.

2 Problem statement

The input to the *perspective generation* task is a sentence s containing a *numeric mention* x: a span of tokens within the sentence which describes a quantity with value $x.\text{value}$ and of unit $x.\text{unit}$. In Figure 1, the numeric mention x is "$131 million", $x.\text{value} = 1.31e8$ and $x.\text{unit} = \$$. The output is a description y that puts x in perspective.

We have access to a knowledge base $\mathcal{K}$ with numeric tuples $t = (t.\text{value}, t.\text{unit}, t.\text{description})$. Table 1 has a few examples of tuples in our knowledge base. Units (e.g. $/per/yr) are fractions composed either of fundamental units (length, area, volume, mass, time) or of ordinal units (e.g. cars, people, etc.).

Description	Value Unit
cost of an employee	71e3 $/year/person
population of Texas	27e3 person
number of employees at Google	57e3 person
average household size	2.54 person
time taken for a basketball game	60 minute
average lifetime for a person	79 year
a week	1 week
time taken for lunch	30 minute
cost of property in the Bay area	1e3 $/ft^2
area of a city block	10e3 m^2

Table 1: A subset of our knowledge base of numeric tuples. Tuples with fractional units (e.g. $/ft^2) can be combined with other tuples to create formulas.

The first step of our task, described in Section 4, is to construct a *formula* f over numeric tuples in $\mathcal{K}$ that has the same value and unit as the numeric mention x. A valid formula comprises of an arbitrary multiplier $f.m$ and a sequence of tuples $f.\text{tuples}$. The value of a formula, $f.\text{value}$, is simply the product of the multiplier and the values of the tuples, and the unit of the formula, $f.\text{unit}$, is the product of the units of the tuples. In Figure 1, the formula has a multiplier of 1 and is composed of tuples $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$; it has a value of $1.3e8$ and a unit of $.

The second step of our task, described in Section 5, is to generate a *perspective* y, a short noun phrase that realizes f. Typically, the utterance will be formed using variations of the descriptions of the tuples in $f.\text{tuples}$.

3 Dataset construction

We break our data collection task into two steps, mirroring formula selection and description generation: first, we collect descriptions of formulas constructed exhaustively from our knowledge base (for generation), and then we use these descriptions to collect preferences for perspectives (for construction).

Collecting the knowledge base. We manually constructed a knowledge base with 142 tuples and 9 fundamental units[1] from the United States Bu-

[1]Namely, length, area, volume, time, weight, money, people, cars and guns. These units were chosen because they

reau of Statistics, the orders of magnitude topic on Wikipedia and other Wikipedia pages. The facts chosen are somewhat crude; for example, though "the cost of an employee" is a very context dependent quantity, we take its value to be the median cost for an employer in the United States, $71,000. Presenting facts at a coarse level of granularity makes them more familiar to the general reader while still being appropriate for perspective generation: the intention is to convey the right scale, not necessarily the precise quantity.

Collecting numeric mentions. We collected 53,946 sentences containing numeric mentions from the newswire section of LDC2011T07 using simple regular expression patterns like `$([0-9]+(,[0-9]+)*(.[0-9]+)? ((hundred)|(thousand)|(million)|(billion)|(trillion)))`. The values and units of the numeric mentions in each sentence were normalized and converted to fundamental units (e.g. from miles to length). We then randomly selected up to 200 mentions of each of the 9 types in bins with boundaries $10^{-3}, 1, 10^3, 10^6, 10^9, 10^{12}$ leading to 4,931 mentions that are stratified by unit and magnitude.[2] Finally, we chose mentions which could be described by at least one numeric expression, resulting in the 2,041 mentions that we use in our experiments (Figure 2). We note that there is a slight bias towards mentions of money and people because these are more common in the news corpus.

Generating formulas. Next, we exhaustively generate valid formulas from our knowledge base. We represent the knowledge base as a graph over units with vertices and edges annotated with tuples (Figure 3). Every vertex in this graph is labeled with a unit u and contains the set of tuples with this unit: $\{t \in \mathcal{K} : t.\text{unit} = u\}$. Additionally, for every vertex in the graph with a unit of the form u_1/u_2, where u_2 has no denominator, we add an edge from u_1/u_2 to u_1, annotated with all tuples of type u_2: in Figure 3 we add an edge from *money/person* to *money* annotated with the three person tuples in Table 1. The set of formulas with unit u is obtained by enumerating all paths in the graph which terminate at the vertex u. The multiplier of the formula is set so that the value of

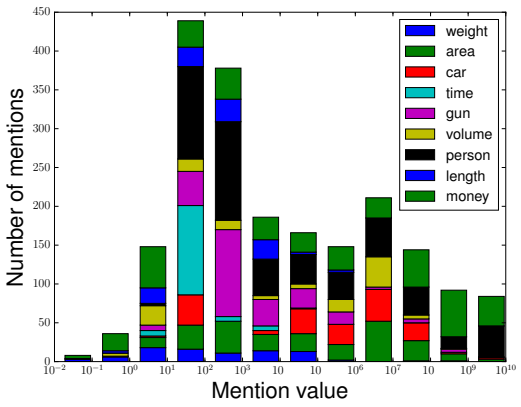

Figure 2: A histogram of the absolute values of numeric mentions by type. There are 100–300 mentions of each unit.

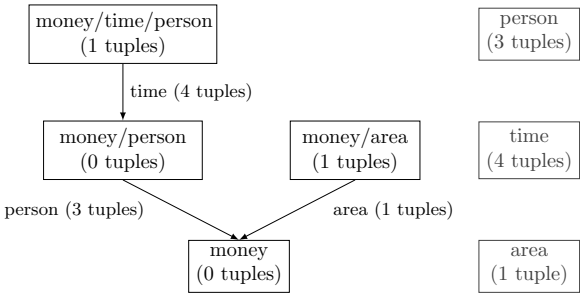

Figure 3: The graph over tuples generated from the knowledge base subset in Table 1.

the formula matches the value of the mention. For example, the formula in Figure 1 was constructed by traversing the graph from *money/time/person* to *money*: we start with a tuple in *money/time/person* (*cost of an employee*) and then multiply by a tuple with unit *time* (*time for lunch*) and then by unit *person* (*population of Texas*), thus traversing two edges to arrive at *money*.

Using the 142 tuples in our knowledge base, we generate a total of 1,124 formulas sans multiplier.

Collecting descriptions of formulas. The main goal of collecting descriptions of formulas is to train a language generation system, though these descriptions will also be useful while collecting training data for formula selection. For every unit in our knowledge base and every value in the set $\{10^{-7}, 10^{-6} \ldots, 10^{10}\}$, we generated all valid formulas. We further restricted this set to formulas with a multiplier between $1/100$ and 100, based on the rationale that human cognition of scale sharply drops beyond an order of magnitude (Tretter et al.,

were well represented in the corpus.

[2]Some types had fewer than 200 mentions for some bins.

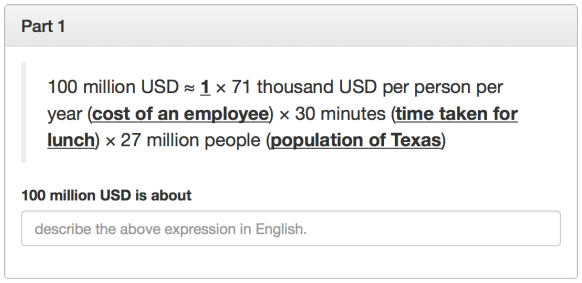

Figure 4: A screenshot of the crowdsourced task to generate natural language descriptions, or perspectives, from formulas.

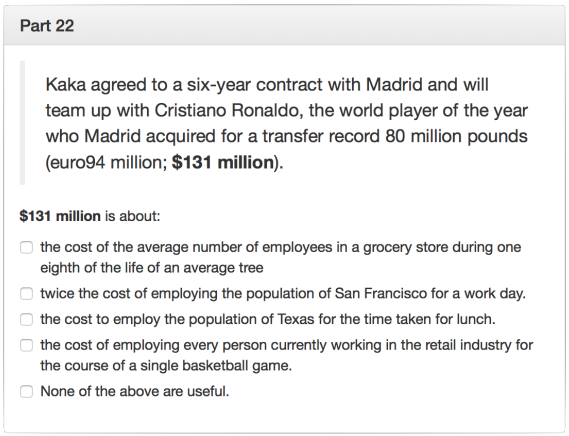

Figure 5: A screenshot of the crowdsourced task to identify which formulas are useful to crowdworkers in understanding the highlighted mentioned number.

2006). In total, 5000 formulas were presented to crowdworkers on Amazon Mechanical Turk, with a prompt asking them to rephrase the formula as an English expression (Figure 4).[3] We obtained 5–7 descriptions of each formula, leading to a total of 31,244 unique descriptions.

Collecting data on formula preference. Finally, given a numeric mention, we ask crowdworkers which perspectives from the description dataset they prefer. Note that formulas generated for a particular mention may differ in multiplier with a formula in the description dataset. We thus relax our constraints on factual accuracy while collecting this formula preference dataset: for each mention x, we choose a random perspective from the description dataset described above corresponding to a formula whose value is within a factor of 2 from the mention's value, $x.\text{value}$. A smaller factor led to too many mentions without a valid comparison, while a larger one led to blatant factual inaccuracies. The perspectives were partitioned into sets of four and displayed to crowdworkers along with a "None of the above" option with the following prompt: "We would like you to pick up to two of these descriptions that are useful in understanding the scale of the highlighted number" (Figure 5). A formula is rated to be useful by simple majority.[4]

Figure 6 provides a summary of the dataset collected, visualizing how many formulas are useful, controlling for the size of the formula. The exhaustive generation procedure produces a large number of spurious formulas like "20 × trash generated in the US × a minute × number of employees on Medicare". Nonetheless, compositional formulas are quite useful in the appropriate context; Table 2 presents some mentions with highly rated perspectives and formulas.

4 Formula selection

We now turn to the first half of our task: given a numeric mention x and a knowledge base $\mathcal{K}$, select a formula f over $\mathcal{K}$ with the same value and unit as the mention. It is easy to generate a very large number of formulas for any mention. For the example, "Cristiano Ronaldo, the player who Madrid acquired for [. . .] $131 million.", the small knowledge base in Table 1 can generate the 12 different formulas,[5] including the following:

1. 1 × the cost of an employee × the population of Texas × the time taken for lunch.
2. 400 × the cost of an employee × average household size × a week.
3. 1 × the cost of an employee × number of employees at Google × a week.
4. 1 × cost of property in the Bay Area × area of a city block.

Some of the formulas above are clearly worse than others: the key challenge is picking a formula that will lead to a meaningful and relevant perspective.

Criteria for ranking formulas. We posit the following principles to guide our choice in features (Table 3).

[3] Crowdworkers were paid $0.08 per description.

[4] Crowdworkers were paid $0.06 to vote on each set of perspectives.

[5] The full knowledge base described in Section 3 can generate 242 formulas with the unit money (sans multiplier).

Sentence	That's about ...	Formula
The Billings-based Stillwater Mining produced **601,000 ounces of platinum**.	4 times the weight of an elephant.	4 × weight of an elephant.
Authorities estimate there are about **60 million guns** in Yemen.	twice the gun ownership of the population of Texas	2 × gun ownership × population of Texas
Water is flowing into Taihu lake at a rate of **150 cubic meters** per second.	how much water would flow from a tap left on for a week.	rate of flow of water from tap × a week
The bank had held auctions, selling around **US$1 billion** worth of three-month bills.	half the cost of employing the population of Texas for a work day.	1/2 × cost of an employee × time taken for a work day × population of Texas
The government[s] have promised to rent about **1.2 million sq. feet**.	the area of forest logged in a single minute	90 × area of forest logged × a minute

Table 2: Examples of numeric mentions, perspectives and their corresponding formulas in the dataset. All the examples except the last one are rated to be useful by crowdworkers.

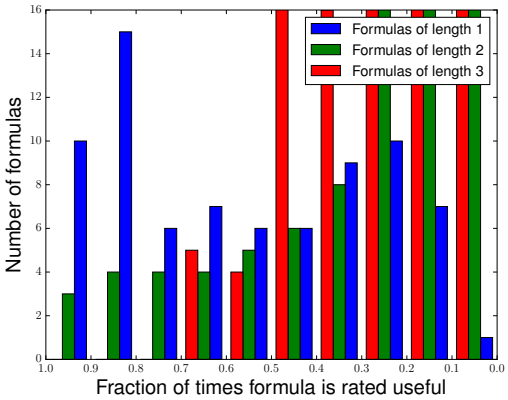

Figure 6: A histogram comparing formula length to ratings of usefulness (clipped for readability). Non-compositional perspectives with a single tuple are broadly useful. Useful compositional perspectives tend to be more context-specific than non-compositional ones, and many of the formulas that can be generated from the knowledge base are spurious.

Type	Features	#
Proximity	$\text{sign}(\log(f.m)), \lvert\log(f.m)\rvert$	1
Familiarity	$\mathbb{I}[t]$	142
Compatibility	$\mathbb{I}[t, t']$	20022
Similarity	$\text{wvec}(s)^\top$ $\text{wvec}(t.\text{description})$	1

Table 3: Feature templates used to score a formulas f and their counts (#), where $f.m$ is the formula's multiplier and $t, t' \in f.\text{tuples}$ are tuples in the formula.

Proximity: *A numeric perspective should be within an order of magnitude of the mentioned value*. Conception of scale quickly fails with quantities that exceed "human scales" (Tretter et al., 2006): numbers that are significantly away from $1/10$ and 10. We use this principle to prune formulas with multipliers not in the range $[1/100, 100]$ (e.g. example 2 above) and introduce features for numeric proximity.

Familiarity: *A numeric perspective should be composed of concepts familiar to the reader*. The most common technique cited by those who do well at scale cognition tests is reasoning in terms of familiar objects (Tretter et al., 2006; Jones and Taylor, 2009; Chevalier et al., 2013). Intuitively, the average American reader may not know exactly how many people are in Texas, but is familiar enough with the quantity to effectively reason using Texas' population as a unit. On the other hand, it is less likely that the same reader is familiar with even the concept of Angola's population.

Of course, because it is so personal, familiarity is difficult to capture. With additional information about the reader, e.g. their location, it is possible to personalize the chosen tuples (Kim et al., 2016). Without this information, we back off to a global preference on tuples by using indicator features for each tuple in the formula.

Formula	Score
Studies estimate **36,000 people** die on average each year from seasonal flu.	
1/4 × global death rate × a day	0.67
5 × death rate in the US × a day	0.64
1/3 × number of employees at Microsoft	0.60
Gazprom's exports to Europe [...] will total **60 billion cubic meters** ...	
oil produced by the US × average lifetime	0.78
average coffee consumption × population of the world × average lifetime	0.78
2 × average coffee consumption × population of Asia × average lifetime	0.73

Table 4: The top three examples outputted by the ranking system with the scores reported by the system.

Compatibility: Similarly, some tuple combinations are more natural ("median income × a month") while others are less so ("weight of a person × population of Texas"). We model compatibility between tuples in a formula using an indicator feature.

Similarity: *A numeric perspective should be relevant to the context*. Apart from helping with scale cognition, a perspective should also place the mentioned quantity in appropriate context: for example, NASA's budget of $17 billion could be described as 0.1% of the United States' budget or the amount of money it could cost to feed Los Angeles for a year. While both perspectives are appropriate, the former is more relevant than the latter.

We model context relevance using word vector similarity between the tuples of the formula and the sentence containing the mention as a proxy for semantic similarity. Word vectors for a sentence or tuple description are computed by taking the mean of the word vectors for every non-stop-word token. The word vectors at the token level are computed using word2vec (Mikolov et al., 2013).

Evaluation. We train a logistic regression classifier using the features described in Table 3 using the perspective ratings collected in Section 3. Recall that the formula for each perspective in the dataset is assigned a positive ("useful") label if it was labeled to be useful to the majority of the workers. Table 5a presents results on classifying formulas as useful with a feature ablation.[6]

Familiarity and compatibility are the most useful features when selecting formulas, each having a significant increase in F_1 over the proximity baseline. There are minor gains from combining these two features. On the other hand, semantic similarity does not affect performance relative to the baseline. We find that this is mainly due to the disproportionate number of unfamiliar formulas present in the dataset that drown out any signal. Table 4 presents two examples of the system's ranking of formulas.

5 Perspective generation

Our next goal is to generate natural language descriptions, also known as perspectives, given a formula. Our approach models the task as a sequence-to-sequence translation task from formulas to natural language. We first describe a rule-based baseline and then describe a recurrent neural network (RNN) with an attention-based copying mechanism (Jia and Liang, 2016).

Baseline. As a simple approach to generate perspectives, we just combine tuples in the formula with the neutral prepositions *of* and *for*, e.g. "1/5th *of* the cost of an employee *for* the population of Texas *for* the time taken for lunch."

Sequence-to-sequence RNN. We use formula-perspective pairs from the dataset to create a sequence-to-sequence task: the input is composed using the formula's multiplier and descriptions of its tuples connected with the symbol '*'; the output is the perspective (Figure 7).

Our system is based on the model described in Jia and Liang (2016). Given a sequence of input tokens ($\mathbf{x} = (x_i)$), the model computes a context-dependent vector ($\mathbf{b} = (b_i)$) for each token using a bidirectional RNN with LSTM units. We then generate the output sequence (y_j) left to right as follows. At each output position, we have a hidden state vector (s_j) which is used to produce an "attention" distribution ($\boldsymbol{\alpha}_j = (\alpha_{ji})$) over input tokens: $\alpha_{ji} = \mathrm{Attend}(s_j, b_i)$. This distribution is used to generate the output token and update the hidden state vector. To generate the token, we ei-

[6] Significance results are computed by the bootstrap test as described in Berg-Kirkpatrick et al. (2012) using the output of classifiers trained on the entire training set.

Feature set	Train			Dev		
	P	R	F_1	P	R	F_1
Proximity	56.4	48.7	52.2	56.3	48.8	52.3
Similarity	65.1	34.9	45.4	65.1	34.9	45.4
Familiarity*	70.5	63.5	66.8	69.6	62.9	66.1
Compatibility$^+$	66.9	**74.4**	70.4	65.4	**73.1**	69.0
F + C†	**73.8**	70.3	**72.1**	**71.5**	68.9	**70.1**
F + C + P†	**73.8**	70.3	**72.1**	**71.5**	68.9	**70.1**
F + C + P + S†	**73.8**	70.3	72.0	71.4	68.6	69.9

(a) the formula construction system. Precision, Recall and F_1 are cross-validated on 10-folds. *significant F_1 versus P and S with $p < 0.01$. $^+$significant F_1 versus P, S and F with $p < 0.01$. †significant F_1 versus P, S, F and C with $p < 0.05$.

System	Train BLEU	Test BLEU
Baseline	65.00	57.32
RNN*	**81.50**	**69.79**

(b) the description generation system. *significant BLEU score versus the baseline with $p < 0.01$.

Table 5: Evaluation of perspective generation subsystems.

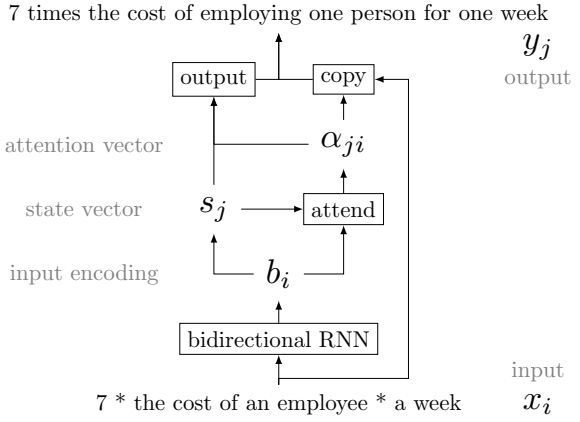

Figure 7: We model description generation as a sequence transduction task, with input as formulas (at bottom) and output as perspectives (at top). We use a RNN with an attention-based copying mechanism.

ther sample a word from the current state or copy a word from the input using attention. Allowing our model to copy from the input is helpful for our task, since many of the entities are repeated verbatim in both input and output. We refer the reader to Jia and Liang (2016) for more details.

Evaluation. We split the perspective description dataset into a training and test set such that no formula in the test set contains the same set of tuples as a formula in the training set.[7] Table 5b compares the performance of the baseline and sequence-to-sequence RNN using BLEU. The sequence-to-sequence RNN performs significantly better than the baseline, producing more natural rephrasings. Table 6 shows some output generated by the system (see Table 6).

6 Human evaluation

In addition to the automatic evaluations for each component of the system, we also ran an end-to-end human evaluation on an independent set of 211 mentions collected using the same methodology described in Section 3. Crowdworkers were asked to choose between perspectives generated by our full system (LR+RNN) and those generated by the baseline of picking the numerically closest tuple in the knowledge base (BASELINE). They could also indicate if either both or none of the shown perspectives appeared useful.[8]

Table 7 summarizes the results of the evaluation and an error analysis conducted by the authors. Errors were characterized as either being errors in generation (e.g. Table 6) or violations of the criteria in selecting good formulas described in Section 4 (Table 7c). The *other* category mostly contains cases where the output generated by LR+RNN appears reasonable by the above criteria but was not chosen by a majority of workers. A few of the mentions shown did not properly describe a numeric quantity, e.g. "... claimed responsibility for a *2009 gun* massacre ..." and were labeled *invalid mentions*. The most common error is the selection of a formula that is not contextually relevant to the mentioned text because no such

[7]Note that formulas with the same set of tuples can occur multiple times in the either the training or test set with different multipliers.

[8]Crowdworkers were paid $0.06 per to choose a perspective for each mention. Each mention and set of perspectives were presented to 5 crowdworkers.

Input formula	Generated perspective
7 × the cost of an employee × a week	7 times the cost of employing one person for one week
1/10 × the cost of an employee × the population of California × the time taken for a football game	one tenth the cost of an employee during a football game by the population of California
1 × coffee consumption × a minute × population of the world	the amount of coffee consumed in one minute on the world
6 × weight of a person × population of California	six times the weight of the people who is worth

Table 6: Examples of perspectives generated by the sequence-to-sequence RNN. The model is able to capture rephrasings of fact descriptions and reordering of the facts. However, it often confuses prepositions and, very rarely, can produce nonsensical utterances.

LR+RNN perspective rated useful?	BASELINE	#
Yes	Yes	31
Yes	No	63
No	Yes	61
No	No	56

(a) A summary of the number of times the perspective generated by LR+RNN or BASELINE was rated useful by a majority of crowdworkers.

Cause of error	#
Proximity	9
Familiarity	6
Compatibility	8
Similarity	49
Generation	24
Other	14
Invalid mention	7
Total	117

(b) An analysis of errors produced by LR+RNN when its perspectives were not rated useful. Errors caused by poor formula selection are further categorized by selection criteria violated.

Cat.	Mention
	LR+RNN perspective (*vs.* BASELINE)
Prox.	... ready to ship about **2,300 miles** across the Pacific to the mainland ...
	three times the distance from San Francisco to Los Angeles (*vs.* the distance from San Francisco to Dallas TX).
Sim.	China had disposed of about **100,000 tons** of CFCs" ...
	one fifth of the weight of garbage produced in the United States by the population of Texas in one week. (*vs.* the average food wasted every year).
Fam.	... the project could save New England ratepayers **$4.6 billion** in energy costs over 25 years.
	one eighth the cost of employing the population of Asia for one hour. (*vs.* the construction cost of The Cosmopolitan in Las Vegas.)
Comp.	Hominids started shaping stone tools about **2.6 million years** ago.
	5 times the total time taken to build the number of cars registered. (*vs.* 17000 times the average lifetime for a tree).

(c) Examples of errors categorized by the criteria defined in Section 4.

Table 7: Results of an end-to-end human evaluation of the output produced by our perspective generation system (LR+RNN) and a baseline (BASELINE) that picks the numerically closest tuple in the knowledge base for each mention.

	Mention	Perspective (that's about...)
+	In 2007, Turkmenistan exported **50 billion cubic meters** of gas to Russia.	the amount of oil produced by the US during a lifetime
+	It can carry up to 10 nuclear warheads and has a range of **8,000 km**.	the distance from San Francisco to Beijing
-	the **2.7 million square feet** that Mission Bay's largest developer is entitled to build	twice the area of forest logged in a minute
-	Las Vegas Sands claims the **10.5 million square feet** is the largest building in Asia.	one half of an area of an average farm

Table 8: Examples of perspectives generated by our system that frame the mentioned quantity to be larger or smaller (top to bottom) than initially the authors thought.

formula exists within the knowledge base (within an order of magnitude of the mentioned value): a larger knowledge base would significantly decrease these errors.

7 Related work and discussion

We have proposed a new task of perspective generation. Compositionality is the key ingredient of our approach, which allows us synthesize information across multiple sources of information. At the same time, compositionality also poses problems for both formula selection and description generation.

On the formula selection side, we must compose facts that make sense. For semantic compatibility between the mention and description, we have relied on simple word vectors (Mikolov et al., 2013), but more sophisticated forms of semantic relations on larger units of text might yield better results (Bowman et al., 2015).

On the description generation side, there is a long line of work in generating natural language descriptions of structured data or logical forms Wong and Mooney (2007); Chen and Mooney (2008); Lu and Ng (2012); Angeli et al. (2010). We lean on the recent developments of neural sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015). Our problem bears some similarity to the semantic parsing work of Wang et al. (2015), who connect generated canonical utterances (representing logical forms) to real utterances.

If we return to our initial goal of helping people understand numbers, there are two important directions to explore. First, we have used a small knowledge base, which limits the coverage of perspectives we can generate. Using Freebase (Bollacker et al., 2008) or even open information extraction (Fader et al., 2011) would dramatically increase the number of facts and therefore the scope of possible perspectives.

Second, while we have focused mostly on basic compatibility, it would be interesting to explore more deeply how the juxtaposition of facts affects framing. Table 8 presents several examples generated by our system that frame the mentioned quantities to be larger or smaller than the authors originally thought. We think perspective generation is an exciting setting to study aspects of numeric framing (Teigen, 2015).

Reproducibility All code, data, and experiments for this paper are available on the CodaLab platform at `https://worksheets.codalab.org/worksheets/0x243284b4d81d4590b46030cdd3b72633/`.

Acknowledgments

We would like to thank Glen Chiacchieri for providing us information about the Dictionary of Numbers, Maneesh Agarwala for useful discussions and references, Robin Jia for sharing code for the sequence-to-sequence RNN, and the anonymous reviewers for their constructive feedback. This work was partially supported by the Sloan Research fellowship to the second author.

References

G. Angeli, P. Liang, and D. Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learn-

ing to align and translate. *arXiv preprint arXiv:1409.0473* .

P. J. Barrio, D. G. Goldstein, and J. M. Hofman. 2016. Improving the comprehension of numbers in the news. In *Conference on Human Factors in Computing Systems (CHI)*.

T. Berg-Kirkpatrick, D. Burkett, and D. Klein. 2012. An empirical investigation of statistical significance in NLP. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 995–1005.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *International Conference on Management of Data (SIGMOD)*. pages 1247–1250.

S. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D. L. Chen and R. J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *International Conference on Machine Learning (ICML)*. pages 128–135.

F. Chevalier, R. Vuillemot, and G. Gali. 2013. Using concrete scales: A practical framework for effective visual depiction of complex measures. *IEEE Transactions on Visualization and Computer Graphics* 19:2426–2435.

G. Chiacchieri. 2013. Dictionary of numbers. `http://www.dictionaryofnumbers.com/`.

A. Fader, S. Soderland, and O. Etzioni. 2011. Identifying relations for open information extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

R. Jia and P. Liang. 2016. Data recombination for neural semantic parsing. In *Association for Computational Linguistics (ACL)*.

M. G. Jones and A. R. Taylor. 2009. Developing a sense of scale: Looking backward. *Journal of Research in Science Teaching* 46:460–475.

Y. Kim, J. Hullman, and M. Agarwala. 2016. Generating personalized spatial analogies for distances and areas. In *Conference on Human Factors in Computing Systems (CHI)*.

W. Lu and H. T. Ng. 2012. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1611–1622.

M. Luong, H. Pham, and C. D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1412–1421.

T. Mikolov, K. Chen, G. Corrado, and Jeffrey. 2013. Efficient estimation of word representations in vector space. *arXiv* .

J. A. Paulos. 1988. *Innumeracy: Mathematical illiteracy and its consequences*. Macmillan.

C. Seife. 2010. *Proofiness: How you're being fooled by the numbers*. Penguin.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. pages 3104–3112.

K. H. Teigen. 2015. Framing of numeric quantities. *The Wiley Blackwell Handbook of Judgment and Decision Making* pages 568–589.

T. R. Tretter, M. G. Jones, and J. Minogue. 2006. Accuracy of scale conceptions in science: Mental maneuverings across many orders of spatial magnitude. *Journal of Research in Science Teaching* 43:1061–1085.

Y. Wang, J. Berant, and P. Liang. 2015. Building a semantic parser overnight. In *Association for Computational Linguistics (ACL)*.

Y. W. Wong and R. J. Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL)*. pages 172–179.