Reasoning about Quantities in Natural Language

Subhro Roy
University of Illinois,
Urbana Champaign
sroy9@illinois.edu

Tim Vieira
Johns Hopkins University
tim.f.vieira@gmail.com

Dan Roth
University of Illinois,
Urbana Champaign
danr@illinois.edu

Abstract

Little work from the Natural Language Processing community has targeted the role of quantities in Natural Language Understanding. This paper takes some key steps towards facilitating reasoning about quantities expressed in natural language. We investigate two different tasks of numerical reasoning. First, we consider Quantity Entailment, a new task formulated to understand the role of quantities in general textual inference tasks. Second, we consider the problem of automatically understanding and solving elementary school math word problems. In order to address these quantitative reasoning problems we first develop a computational approach which we show to successfully recognize and normalize textual expressions of quantities. We then use these capabilities to further develop algorithms to assist reasoning in the context of the aforementioned tasks.

1 Introduction

Every day, newspaper articles report statistics to present an objective assessment of the situations they describe. From election results, number of casualties in accidents, to changes in stock prices, textual representations of quantities are extremely important in communicating accurate information. However, relatively little work in Natural Language Processing has analyzed the use of quantities in text. Even in areas where we have relatively mature solutions, like search, we fail to deal with quantities; for example, one cannot search the financial media for "transactions in the 1-2 million pounds range."

Language understanding often requires the ability to reason with respect to quantities. Consider, for example, the following textual inference, which we present as Textual Entailment query. Recognizing Textual Entailment (RTE) (Dagan et al., 2013) has become a common way to formulate textual inference and we follow this trend. RTE is the task of determining whether the meaning of a given text passage T entails that of a hypothesis H.

Example 1
T:*A bomb in a Hebrew University cafeteria killed five Americans and four Israelis.*
H:*A bombing at Hebrew University in Jerusalem killed nine people, including five Americans.*

Here, we need to identify the quantities *"five Americans"* and *"four Israelis"*, as well as use the fact that *"Americans"* and *"Israelis"* are *"people"*.

A different flavour of numeric reasoning is required in math word problems. For example, in one has to determine the relevant quantities in the question. Here, the number of blocks in Ryan's possession has no bearing on the answer. The second challenge is to determine the relevant mathematical operation from the context.

Example 2
Ryan has 72 *marbles and* 17 *blocks. If he shares the marbles among* 9 *friends, how many marbles does each friend get?*

In this paper, we describe some key steps necessary to facilitate reasoning about quantities in natural language text. We first describe a system developed to recognize quantities in free form text, infer units associated with them and convert them to

Transactions of the Association for Computational Linguistics, vol. 3, pp. 1–13, 2015. Action Editors: Johan Bos, Lillian Lee.
Submission batch: 6/2014; Revision batch 9/2014; Published 1/2015.

a standardized form. For example, in

Example 3
About six and a half hours later , Mr. Armstrong opened the landing craft's hatch.

we would like to extract the number 6.5, the corresponding unit, *"hour"*, and also determine that the quantity describes an *approximate* figure, not an exact one. One of the difficulties is that any noun or noun phrase can be a unit, and inferring them requires analyzing contextual cues and local sentence structure. As we show, in some cases deeper NLP techniques are required to support that.

We then develop a reasoning framework for quantities that we believe can play an important role in general purpose textual inference. Isolating the quantity reasoning component of the RTE task, we define Quantity Entailment (QE) - the task of determining whether a given quantity can be inferred from a given text snippet, and then describe our approach towards solving it. This allows us to support the inference presented in Example 1.

As an additional evaluation, we also show the effectiveness of our system on an application of QE, a search for ranges of currency values. Given a query range, say from 1 million USD to 3 million USD, we want to find all mentions of money with values in this range. Using standard search engine technology to query all values in the range, in the various forms they could be expressed, is not feasible. Instead, we use our proposed approach to extract monetary mentions from text and normalize them, and then we use QE to verify them against the query.

We next develop a reasoning framework for elementary school math word problems. Our reasoner makes use of several classifiers to detect different properties of a word problem, and finally combines the decisions of individual classifiers to obtain the correct answer.

We develop and annotate datasets[1] for evaluation, and show that our approach can handle the aforementioned reasoning tasks quite well.

The next section presents some related work on quantities and reasoning. We then formally define a *quantity* and describe our knowledge representation. The following sections describe quantities extraction and standardization. We next present the formulation of Quantity Entailment, and describe our reasoning framework for it. We then describe our approach towards understanding elementary school math problems, and conclude with experimental evaluation.

2 Related Work

The importance of reasoning about quantities has been recognized and studied from multiple perspectives. Quantities have been recognized as an important part of a textual entailment system (de Marneffe et al., 2008; Maccartney and Manning, 2008; Garoufi, 2007; Sammons et al., 2010), and (de Marneffe et al., 2008) claims that discrepancies in numbers are a common source of contradictions in natural language text. The authors describe a corpus of real-life contradictory pairs from multiple sources such as Wikipedia and Google News in which they found that 29% of the contradictions were due to numeric discrepancies. In addition, they analyzed several Textual Entailment datasets (Dagan et al., 2006) and found that numeric contradictions constitute 8.8% of contradictory entailment pairs.

Quantitative reasoning has also been addressed from the perspective of formal semantics. Montague (Montague, 1973) investigates identity ambiguities in sentences, e.g., whether *"The temperature is ninety but it is rising."* implies *"ninety is rising"*. His solution suggests that *"temperature"* should be treated as a concept, and *"temperature is ninety"* asserts an attribute of temperature at a particular instance of time, and not an attribute of the concept *"temperature"*. Reasoning about quantities often depends on reasoning about monotonicity. The role of monotonicity in NL reasoning has been described in (Barwise and Cooper, 1981). The authors categorize noun phrases as upward or downward monotonic, and also detect constructs where monotonicity depends on context. The large role of monotonicity in reasoning motivated attempts to reason directly at the surface level (Purdy, 1991), rather than converting first to logical forms. Our approach advocates this direction too.

(Kuehne, 2004a) investigates the various cases in which physical quantities are represented

[1] The datasets are available for download at http://cogcomp.cs.illinois.edu/page/resource_view/95.
The related software is available at http://cogcomp.cs.illinois.edu/page/software_view/Quantifier.

in descriptions of physical processes. Later, in (Kuehne, 2004b), a system to extract Qualitative Process Theory (Forbus, 1984) representations is implemented for a controlled subset of the English language. Other works that are relevant to quantities, such as work on the plural semantics of noun phrases (Schwertel, 2003), were also done on controlled English. While these approaches do not scale to unrestricted English, they have influenced the quantity representation that we use.

The importance of quantities has also been recognized in some application areas. For example, (Banerjee et al., 2009) investigates ranking of search results involving quantities. In order to detect quantities in text, they use a rule based system, comprising 150 rules. However, the rules were specific to the queries used, and do not extend well to unrestricted English. In contrast, our system is designed to detect any quantity mentioned in natural language text, as well as infer the unit associated with it. There has also been some work on quantities in specific domains, such as the temporal domain, the most significant being the TimeML project (Pustejovsky et al., 2003; Saur et al., 2005; Pratt-Hartmann, 2005; Do et al., 2012). The problem of automatically solving math word problems has also been investigated. Approaches range from using rule-based methods (Bobrow, 1964; Lev et al., 2004; Mukherjee and Garain, 2008) to recent template matching techniques (Kushman et al., 2014) .

3 Representing Quantities

In general, quantity refers to anything which is measurable. Our quantities representation is influenced by the one proposed in (Forbus, 1984) but we propose a simpler version of their Qualitative Process theory:

Definition (Quantity-Value Representation) In Quantity-Value Representation (QVR), a quantity is represented as a triple (v, u, c), where constituents in the triple correspond, respectively, to:

1. Value: a numeric value, range, or set of values which measure the aspect, e.g. more than 500, one or two, thousands, March 18, 1986. The value can also be described via symbolic value (e.g., "below the freezing point"). We do not store surface forms explicitly, but convert them to a set or range. For example, "more than 500" is stored as the range $(500, +\infty)$. Details of these conversions are given in Section 4.2.
2. Units: a noun phrase that describes what the value is associated with. e.g., inches, minutes, bananas. The phrase "US soldiers" in the phrase "Five US soldiers" is a unit.
3. Change: specifies how the parameter is changing, e.g., increasing. This constituent often serves as an indication of whether or not the value is relative to another. For example, "She will receive an [additional 50 cents per hour]", "The stock [increased 10 percent]", "Jim has [5 balls more] than Tim".

4 Extraction of Quantities

In this section we describe the first component of our approach, that of identifying quantities and units in text and standardizing their representation. We use a a two step approach to extract quantities from free form text.

1. **Segmentation** This step takes raw text and finds segments of contiguous text which describe quantities.
2. **Standardization** Using the phrases extracted in the previous step, we derive the QVR.

An overview of our method is given in Algorithm 1.

Algorithm 1 QuantityExtraction(T)

```
Input: Text T
Output: Set of Quantity-value triples extracted from T
$Q \leftarrow \emptyset$
$S \leftarrow$ Segmentation( T )
for all segment $s \in S$ do
    $q \leftarrow$ Standardization( s )
    if unit of $q$ not inferred then
        $q \leftarrow$ InferUnitFromSemantics( $q$, $s$, T )
    end if
    $Q \leftarrow Q \cup \{q\}$
end for
return $Q$
```

We model the **segmentation** step as a sequence segmentation task because quantities often appear as segments of contiguous text. We adapt and compare two approaches that were found successful in previous sequential segmentation work in NLP:

1. A Semi-CRF model (Sarawagi and Cohen, 2004), trained using a structured Perceptron algorithm (Collins, 2002), with Parameter Averaging (Freund and Schapire, 1998).
2. A bank of classifiers approach (Punyakanok and Roth, 2001) that we retrain with a new set of features.

The same feature set was used for both approaches. Despite the additional expressive power of CRFs, we found that the bank of classifiers (which is followed by a simple and tractable inference step) performs better for our task, and also requires significantly less computation time.

4.1 Features

For each token x_i in the input sequence we extract the following features:

1. **Word class features**: x_i appears in a list of known scientific units (e.g., meters, Fahrenheit), written numbers (e.g., two, fifteen), names of a months, day of the week, miscellaneous temporal words (e.g. today, tomorrow), currency units, etc.
2. **Character-based**: x_i contains a digit, is all digits, has a suffix (st,nd,rd,th).
3. **Part of speech tags**: we use the Illinois POS Tagger (Roth and Zelenko, 1998).
4. Most of the features were generated from a window of $[-3, 3]$ around the current word. Additional features were generated from these by conjoining them with offset values from the current word.

4.2 Mapping Text Segments into QVR

We develop a rule-based **standardization** step, that is informed, as needed, by deeper NL processing, including semantic role labeling (SRL, (Palmer et al., 2010)) and Co-reference resolution. Some key steps of this procedure are as follows:

1. Convert written numbers to floating point: e.g., three thousand five hundred twenty → 3520.0
2. Convert dates to an internal date type: e.g., March 18th → $\mathrm{Date}(03/18/\mathrm{XXXX})$
3. Replace known names for ranges: e.g., teenage → $[13, 19]$ years-old.
4. Convert all scientific units to a standard base unit: e.g., 1 mile → 1609.344 meters.
5. Replace non-scientific units with WordNet synsets
6. Rewrite known units to a standard unit: e.g., USD, US$, dollars → US$.
7. Standardize changing quantity: e.g., “additional 10 books” → $+10$ [book].
8. Extract bounds: we use a list of phrases, such as “more than”, “less than”, “roughly”, “nearly”. By default, if a bound keyword is not present we assume the bound is “=”.
9. Modify value using bounds : We convert values which have a bound to a range of values.

   Scalar implicature is taken into consideration here. Consider the sentence “John bought 10 books.”, although it can be interpreted that buying 5 books is a corollary of buying 10, in this case, we make the assumption that 5 books were not purchased. See section 5.2 for a discussion on the subject.

   We use the following rules, where v is the value extracted before using bound information.

   - $\leq v \rightarrow (-\infty, v]$, similarly for $\geq, <, >$.
   - $= v \rightarrow \{v\}$
   - $\approx v \rightarrow [v - c.v, v + c.v]$, we use $c = 0.2$.

4.3 Extraction of Units

In most cases, the units related to the numeric values appear adjacent to them. For example, in the sentence *“There are two books on the table”*, the unit *“book”* follows *“two”*. The sequence segmentation groups these words together, from which it is easy to extract the unit. However, in some cases, a better understanding of the text is needed to infer the units. Consider the following example:

Example 4
A report from UNAIDS, the Joint United Nations Program on HIV/AIDS, released on Tuesday, shows the number of adults and children with HIV/AIDS reached 39.4 million in 2004.

Here, we need to know that *"39.4 million"* refers to *"the number of adults and children with HIV/AIDS"*. Also, in:

Example 5
The number of member nations was 80 in 2000, and then it increased to 95.

we need to know that the pronoun *"it"* refers to *"the number of member nations"*.

We employ a sequential process in our standardization. In case the first step described above fails to extract units, we make use of deeper processing of the sentence to accomplish that (see an evaluation of the contribution of this in the experimental section). These steps are denoted by the function **InferUnitFromSemantics()** in Algorithm 1. We apply coreference resolution to identify pronoun referents and then apply a Semantic Role Labeler, to recognize which terms are associated with the quantity, and can be potential units. In the case of example 4, the SRL tells us that for the verb *"reached"*, the associated subject is *"the number of adults and children with HIV/AIDS"* and the object is the mention *"39.4 million"*. Hence, we conclude that the subject can be a candidate for the unit of *"39.4 million"*. For the purpose of entailment, we keep the entire set of possible word chunks, which are linked by the SRL to our quantity mention, as candidate units.

Since most units are found in positions adjacent to the numeric mention, we optimize on runtime by applying the SRL and coreference resolver only when the segmented chunk does not have adequate information to infer the unit. We use the Illinois Coreference Resolver (Bengtson and Roth, 2008; Chang et al., 2013) and the Illinois SRL (Punyakanok et al., 2008), for coreference and semantic role labelling, respectively.

5 Quantity Entailment

In this section we describe our approach to quantitative reasoning from natural language text. We first formulate the task of Quantity Entailment, and then describe our reasoning framework.

Definition (Quantity Entailment) Given a text passage T and a Quantity-Value triple $h(c_h, v_h, u_h)$, Quantity Entailment is a 3-way decision problem:

1. **entails**: there exists a quantity in T which entails h.
2. **contradicts**: no quantity in T entails h, but there is a quantity in T which contradicts h.
3. **no relation**: there exists no quantity in T, which is comparable with h.

The need to identify sub-problems of textual inference, in the context of the RTE task, has been motivated by (Sammons et al., 2010). Quantity Entailment can be considered as one such step. Since we envision that our QE module will be one module in an RTE system, we expect that the RTE system will provide it with some control information. For example, it is often important to know whether the quantity is mentioned in an upward or downward monotonic context. Since we are evaluating our QE approach in isolation, we will always assume upward monotonicity, which is a lot more common. Monotonicity has been modeled with some success in entailment systems (Maccartney and Manning, 2008), thus providing a clear and intuitive framework for incorporating an inference resource like the Quantity Entailment module into a full textual entailment system.

5.1 Reasoning Framework

Our Quantity Entailment process has two phases: Extraction and Reasoning. In the Extraction Phase, we take a text passage T and extract Quantity-Value triples (value, units, change) from it. In the Reasoning phase, we apply a lightweight logical inference procedure to the triples extracted from T to check if h can be derived.

There are two types of rules applied in the Reasoning phase: Implicit Quantity Productions and Quantity Comparisons. The combination of these rules provides good coverage for the QE task.

5.1.1 Quantity Comparison

Quantity Comparison compares a quantity $t : (v_t, u_t, c_t)$ extracted from T and the quantity $h : (v_h, u_h, c_h)$ and decides whether h can be derived via some truth preserving transformation of t. There are three possibilities: (t entails h), (t contradicts h), or (t has no relation with h). The overview is given in Alg. 2, which is designed under our assumption that entailing quantities should respect upward monotonicity. This requires monotonicity verification of both units and values.

In order for a quantity to contradict or entail another, their units must be comparable. Determining the comparability of scientific units is direct since they form a closed set. Comparing non-scientific units is more involved. The inference rule used here is as follows: if the syntactic heads of the unit phrases match (i.e., there is an Is-A or synonymy relation in either direction), then the phrases are comparable. These comparisons are encoded as a function **comparableUnits**(u_t, u_h), which returns true if the units u_t and u_h are comparable, or else returns false.

If the units are comparable, the direction of monotonicity (i.e., the direction of the Is-A relation between the heads and the effects of any relevant modifiers) is verified. The function **checkMonotonicityOfUnits**(u_t, u_h) returns true, if u_t is more specific than u_h, false otherwise.

To compute the Is-A and synonymy relations we use WordNet (Miller et al., 1990), an ontology of words which contains these relations. We also augment WordNet with two lists from Wikipedia (specifically, lists of Nationalities and Jobs).

Next, we check whether the values of the quantities compared obey the monotonicity assumption; we say that v_t is more specific than v_h if v_t is a subset of v_h. (Note that v_t and v_h are both represented as sets and hence, checking subset relation is straightforward.) For example, *"more than 50"* $\subseteq$ *"at least 10"*. This rule also applies to dates, e.g. *"03/18/1986"* $\subseteq$ *"March 1986"*. Respecting scalar implicature, we assume that "5" is subset of "less than 10", but not "10". Similar to the case of units, we use the function **checkMonotonicityOfValues**(v_t, v_h) which returns true, if v_t is more specific than v_h, and false otherwise.

A quantity which represents some form of change of a quantity cannot be derived from a quantity which does not represent change and vice versa. We set c_t = true if t denotes change in a quantity, otherwise we set c_t = false.

Algorithm 2 QuantityComparison(t, h)

```
Input: Quantity-value triples $t(v_t, u_t, c_t)$ and $h(v_h, u_h, c_h)$
Output: Returns whether $t$ entails, contradicts or has no relation with $h$
if $c_t \neq c_h$ then
    return no relation
end if
if comparableUnits( $u_t$, $u_h$ )= false then
    return no relation
end if
if checkMonotonicityOfUnits( $u_t$, $u_h$ )= true then
    if checkMonotonicityOfValues( $v_t$, $v_h$ )= true then
        return entails
    end if
end if
return contradicts
```

5.1.2 Implicit Quantity Production Rules

There are many relationships among quantities which can be the source of implicit information. The following is an incomplete, but relatively broad coverage list of common patterns:

1. Range may imply duration, e.g., *"John lived in Miami from 1980 to 2000"* implies that John lived in Miami for a duration of 20 years.

2. Compatible terms may be combined and abstracted. The sentence *"I bought 3 bananas, 2 oranges, and 1 apple"* implies that 6 fruits were purchased.

3. Ratios can imply percentages. The sentence *"9 out of the 10 dentists interviewed recommend brushing your teeth"* implies that 90% of the dentists interviewed recommend brushing.

4. Composition: Quantities and units may sometimes be composed. Consider the following examples, the phrase *"six Korean couples"* means that there are 12 people; the phrase *"John gave six 30-minute speeches"* implies that John spoke for 180 minutes.

The rules used for producing implicit quantities employed in our system are the following:

- **(a ratio b)** if a is a percentage, then multiply its value with the value of b to obtain a new quantity with the units of b.

- **(a ratio b)** if a is not percentage, divide its value with the value of b to obtain a new quantity with the units of b.

- **(a range b)** take the difference of the two values to obtain a new quantity with the appropriate change of units, e.g., time-stamp minus time-stamp results in units of time.

Algorithm 3 QuantityEntailment(T, h)

```
Input: Text T and a quantity-value triples $h(v_h, u_h, c_h)$
Output: Returns whether T *entails*, *contradicts* or has *no relation* with $h$
$Q \leftarrow$ QuantityExtraction( T )
$Q' \leftarrow$ GenerateImplicitQuantities( $Q$ )
$Q \leftarrow Q \cup Q'$
contradict $\leftarrow$ false
for all quantity-value triple $q \in Q$ do
    if QuantityComparison( $q$, $h$ )= *entails* then
        return entails
    end if
    if QuantityComparison( $q$, $h$ )= *contradicts* then
        contradict$\leftarrow$ true
    end if
end for
if contradict= true then
    return *contradicts*
else
    return *no relation*
end if
```

5.1.3 Lightweight Logical Inference

The QE inference procedure simply applies each of the implicit quantity production rules to the Quantity-Value triples extracted from the passage T, until no more quantities are produced. Then it compares each quantity t extracted from T with the quantity h, according to the quantity comparison rules described in Algorithm 2. If any quantity in T entails h, then "entails" is reported; if there is no quantity in T which can explain h, but there exists one which contradicts h, then "contradiction" is reported; otherwise "no relation" is reported. The complete approach to Quantity Entailment is given in Algorithm 3.

5.2 Scope of QE Inference

Our current QE procedure is limited in several ways. In all cases, we attribute these limitations to subtle and deeper language understanding, which we delegate to the application module that will use our QE procedure as a subroutine. Consider the following examples:

T : Adam has exactly 100 dollars in the bank.
H_1 : Adam has 50 dollars in the bank.
H_2 : Adam's bank balance is 50 dollars.

Here, T implies H_1 but not H_2. However for both H_1 and H_2, QE will infer that "50 dollars" is a contradiction to sentence T, since it cannot make the subtle distinction required here.

T : Ten students passed the exam, but six students failed it.
H : At least eight students failed the exam.

Here again, QE will only output that T implies "At least eight students", despite the second part of T. QE reasons about the quantities, and there needs to be an application specific module that understands which quantity is related to the predicate "failed".

There also exists limitations regarding inferences with respect to events that could occur over a period of time. In *"It was raining from 5 pm to 7 pm"* one needs to infer that *"It was raining at 6 pm"* although *"6 pm"* is more specific than *"5 pm to 7 pm"*. There is a need to understand the role of associated verbs and entities, and the monotonicity of the passages to infer the global entailment decision. Some aspects of this problem is handled in the math word problems in Section 6, but there is still a need to formalize the role of associated predicates and its associations with quantities in natural language.

6 Solving Math Word Problems

In this section, we describe our approach towards automatically understanding and solving elementary school math word problems. We considered word problems having the following properties:

1. The question mentions two or three quantities.

2. The answer can be computed by choosing

two quantities from the question and applying one of the four basic operations (addition, subtraction, multiplication, division) on them.

We use a cascade of classifiers approach for this problem. We develop the following three classifiers to detect different properties of the word problem.

1. **Quantity Pair Classifier** This classifier is relevant only for problems mentioning three quantities in the question text. The input to the classifier is the text of the question Q of the problem, and the quantities q_1, q_2, q_3 extracted from the question Q. The output is the relevant pair of quantities, that is, the pair of quantities required to get the answer, denoted as (q_i, q_j). The inference problem can be written as follows:

   $$(q_i, q_j) \leftarrow \arg\max_{p \in P} w_{qp}^T \phi_{qp}(Q, p)$$

   where $P = \{(q_1, q_2), (q_2, q_3), (q_3, q_1)\}$, $\phi_{qp}(\cdot)$ is a feature function, and w_{qp} is a learned weight vector.

2. **Operation Classifier** This classifier takes as input the question Q of the problem, and the relevant quantity pair (q_i, q_j) (decided by Quantity Pair Classifier in case of questions with three quantities), and outputs which of the four operations is required for the problem. The inference in this case is

   $$op \leftarrow \arg\max_{op \in O} w_{opr}^T \phi_{opr}(Q, (q_i, q_j), op)$$

   where $O = \{+, -, \times, /\}$.

3. **Order Classifier** This classifier is relevant only for problems which require subtraction or division. It takes as input the question Q of the problem, the relevant pair of quantities (q_i, q_j) and the operation *op* being performed, and decides the most likely order of quantities in the operation, that is, whether we should perform (q_i op q_j) or (q_j op q_i). The inference can be written as

   $$(q_i', q_j') \leftarrow \arg\max_{p \in P} w_{or}^T \phi_{or}(Q, (q_i, q_j), op, p)$$

   where $P = \{(q_i, q_j), (q_j, q_i)\}$

Algorithm 4 SolveWordProblem(Q)

```
Input: Text of question $Q$
Output: Returns answer to question $Q$
$(q_1, q_2, q_3) \leftarrow$ QuantityExtraction( $Q$ )
$(q_i, q_j) \leftarrow$ QuantityPairClassifier($Q$)
$op \leftarrow$ OperationClassifier($Q$,$(q_i, q_j)$)
$(q_i', q_j') \leftarrow$ OrderClassifier($Q$,$(q_i, q_j)$,$op$)
return ($q_i'$ op $q_j'$)
```

The inference procedure is given in Algorithm 4. For our classifiers, we use a sparse averaged perceptron implemented with the SNOW framework (Carlson et al., 1999). Each classifier is trained on gold annotations for that particular task. The features used are as follows:

1. Unigrams and bigrams from sentences containing quantities.
2. POS tags from sentences with quantities.
3. Relevant pair of quantities, and whether their units match and whether their units are present in the last sentence of the question.
4. Relevant operation for the problem (for Operation and Order classifiers)
5. Relevant order of quantities for the operation (for Order classifier).
6. Various conjunctions of the above features.

7 Experimental Study

In this section, we seek to validate our proposed modeling. We evaluate our system's performance on four tasks: Quantity Segmentation, Quantity Entailment, Currency Range Search, and Answering Math Word Problems. We do not directly evaluate our system's ability to map raw text segments into our representation, but instead evaluate this capability extrinsically, in the context of the aforementioned tasks, since good Standardization is necessary to perform quantitative inference.

7.1 Datasets

QE: Due to lack of related work, an adequately annotated corpus does not exist. Thus, in order to evaluate our system, we used two collections:

1. **Sub-corpus of the RTE Datasets** (Dagan et al., 2006) We choose text-hypothesis pairs from RTE2–RTE4 datasets, which have quantity mentions in the hypothesis. Overall, we selected 384 text-hypothesis pairs with quantities in the hypothesis.

2. **Newswire Text** 600 sentences of newswire text were selected, all containing quantity mentions.

Both these datasets were manually annotated with the phrase boundaries of quantity mentions and had an inter-annotator agreement of 0.91. We restricted annotation to contiguous segments of text. No instances of implicit quantities were annotated. We also did not annotate these mentions with QVRs.

Limiting the annotations to contiguous spans of text results in a few instances of quantities which contain missing information, such as missing or ambiguous units, and several range and ratio relationships which were not annotated (e.g., we do not annotate the range expressed in "*from [5 million] in [1995] to [6 million] in [1996]*", but do so in "*[from 5 million to 6 million]*").

In the RTE sub-corpus we also annotated entailment pairs with information about which quantities entail, in addition to the boundary information. For each quantity in the hypothesis we labeled it as either "entails", "no relation", or "contradicts", with an inter-annotator agreement of 0.95. There were 309 entailing quantities, 71 contradicting quantities and 56 quantities which were unrelated to the corresponding text. We also maintained the information about general entailment, that is, whether the hypothesis can be explained by the text. An example of an annotated RTE example is shown below.

Annotation Example for RTE sub-corpus
T:*A bomb in a Hebrew University cafeteria killed [five Americans] and [four Israelis].*
H:*A bombing at Hebrew University in Jerusalem killed [nine people], including [five Americans].*

"nine people" : entails
"five Americans" : entails
Global entailment decision : entails

Although we limit our scope to infer the entailment decision for individual quantities mentioned in hypothesis, we hope to see future approaches use these individual decisions and combine them appropriately to obtain the global entailment decision.

Currency Search We developed a new dataset for evaluating currency search. Queries of various amounts of money like "1000$", "USD 2 million", etc. were made on a search engine, and paragraphs containing monetary mentions were taken from the top search results. We collected 100 paragraphs containing various mentions of monetary values, and labeled them with the amount mentioned in them. We restricted the denominations to US dollars. The inter-annotator agreement was 0.98.

Math Word Problems We created a new dataset with elementary math word problems. The problems were collected from http://www.k5learning.com/ and http://www.dadsworksheets.com/. The list was further pruned to keep problems with the properties listed in section 6. We also manually removed problems requiring background knowledge, for example, *"Roger reads 2 books each day. How many books will he read in 3 weeks ?"*, which requires knowing that a week comprises 7 days. Problems with rounding issues were also excluded. For example, *"Each basket can hold 9 apples. How many baskets are required to hold 10 apples ?"*. Each problem was annotated with the operation required to solve the problem, and the final answer. Table 1 shows some statistics of our dataset.

#quantities	Relevant Operation			
	Add	Subtract	Multiply	Divide
2	228	214	257	260
3	107	132	75	131

Table 1: Statistics of math word problems dataset

7.2 Quantity Segmentation

We evaluate the phrase boundary recognizer on the annotated RTE and newswire datasets described in the previous section, using the phrase-based F_1 score. We compare the accuracy and running times of the Semi-CRF model (SC) (Sarawagi and Cohen, 2004) and the bank of classifiers model (C+I) (PR) (Punyakanok and Roth, 2001), using 10-fold cross-validation. Note that the standardizer can often recover from mistakes made at the segmentation level. Therefore, this performance does not necessarily upper bound the performance of the next step in our pipeline.

The segmentation we are aiming for does not directly follow from syntactic structure of a sentence. For example, in the sentence *“ The unemployment rate increased 10%”*, we would like to segment together *“increased 10%”*, since this tells us that the quantity denotes a rise in value. Also, in the sentence *“Apple restores push email in Germany, nearly two years after Motorola shut it down”* we would like to segment together *”nearly two years after”* . We consider a quantity to be correctly detected only when we have the exact phrase that we want, otherwise we consider the segment to be undetected.

Model	P%	R%	F%	Train Time	Test Time
Semi-CRF (SC)	75.6	77.7	76.6	15.8	1.5
C+I (PR)	80.3	79.3	79.8	1.0	1.0

Table 2: 10-fold cross-validation results of segmentation accuracy and time required for segmentation, the columns for runtime have been normalized and expressed as ratios

Table 2 describes the segmentation accuracy, as well as the ratio between the time taken by both approaches. The bank of classifiers approach gives slightly better accuracy than the semi-CRF model, and is also significantly faster.

7.3 Quantity Entailment

We evaluate the complete Quantity Entailment system, determining the overall loss due to the segmentation, as well as the contribution of the Coreference Resolver and SRL. We show the performance of 4 systems.

1. GoldSeg : Uses gold segmentation, and does not use SRL and Coreference Resolver.
2. GoldSeg+Sem : Uses gold segmentation, and also uses SRL and Coreference Resolver to infer units.
3. PredSeg : Performs segmentation, and does not use SRL and Coreference Resolver.
4. PredSeg+Sem : Performs segmentation, and uses SRL and Coreference Resolver.

The baseline is an exact string matching algorithm. It answers “entails” if the quantity unit and value are present in the text, and answers “contradicts” if only the unit matches and the value does not. Otherwise, it returns “no relation”. The results are shown in Table 3. Note that exact match only supports 43.3% of the entailment decisions. It is also evident that the deeper semantic analysis using SRL and Coreference improves the quantitative inference.

Task	System	P%	R%	F%
Entailment	Baseline	100.0	43.3	60.5
	GoldSeg	98.5	88.0	92.9
	+Sem	97.8	88.6	93.0
	PredSeg	94.9	76.2	84.5
	+Sem	95.4	78.3	86.0
Contradiction	Baseline	16.6	48.5	24.8
	GoldSeg	61.6	92.9	74.2
	+Sem	64.3	91.5	75.5
	PredSeg	51.9	79.7	62.8
	+Sem	52.8	81.1	64.0
No Relation	Baseline	41.8	71.9	52.9
	GoldSeg	81.1	76.7	78.8
	+Sem	80.0	78.5	79.3
	PredSeg	54.0	75.4	62.9
	+Sem	56.3	72.7	63.5

Table 3: Results of QE; Adding Semantics(+SEM) consistently improves performance; Only 43.3% of entailing quantities can be recovered by simple string matching

7.4 Currency Range Search

Table 4 shows the performance of our system in detecting currency phrases. We evaluate our system on the proportion of monetary mentions it recognized and standardized correctly from queried ranges of currency values, and report micro-averaged scores. Note that range search is a direct application of QE, where the quantity is a range of values, and the text is the corpus we want to search. All instances of “entails” correspond to search hits. The baseline here is also a string matching algorithm, which searches for numbers in the text.

System	P%	R%	F%
Baseline	72.0	69.2	70.5
PredSeg+Sem	96.0	93.5	94.8

Table 4: Micro-averaged accuracy in detecting monetary mentions

7.5 Elementary Math Word Problems

Table 5 shows the performance of individual classifiers as well as the ability of our system to answer correctly math word problems, using the output of the classifiers. The results are reported with respect to 2-fold cross-validation. The accuracy of each classifier is based only on the relevant examples for that particular classifier. For example, Quantity Pair classifier is evaluated on problems with three quantities in its question text, and Order classifier is evaluated on problems concerning subtraction or division. Correct Answer denotes the end to end system, which outputs the answer, after receiving as input the question text of the problem.

Module	Accuracy
Quantity Pair	94.3
Operation	91.8
Order	95.9
Correct Answer	86.9

Table 5: 2-fold cross-validation results of math word problem understanding. Correct Answer indicates performance of end to end system, others represent individual classifier performance

We find that the individual classifiers have high accuracy, and hence our system performs well on the end to end task. A potential future direction can be to propagate the uncertainty in each classifier, which might further improve performance of the system.

7.6 Qualitative Analysis

The segmentation module made mistakes in detecting exact boundaries for uncommon phrases, e.g., “hundreds of thousands of people”, and “mid-1970’s”. Detection of missing units is problematic in cases like “Three eggs are better than two”. The SRL returns “Three eggs” as a candidate unit, which needs to be pruned appropriately to obtain the correct unit. The primary limitation of the reasoning system in both tasks is the lack of an extensive knowledge base. Wordnet based synsets prove to be insufficient to infer whether units are compatible. Also, there are certain reasoning patterns and various implicit relations between quantities which are not currently handled in the system. For example, inferring from the sentence *“Militants in Rwanda killed an [average of 8,000 people per day] for [100 days]”* that “around 800,000 people were killed”. Also, implication of ratios can be involved. For example, the sentence *“[One out of 100 participating students] will get the award”* implies that there were *“100 participating students”*, whereas *“[9 out of 10 dentists] recommend brushing”* does not imply there were 10 dentists. In case of word problems, our system missed non-standard questioning patterns with involved reasoning. For example, *“Bryan has 50 skittles. Ben has 20 M&Ms. Who has more? How many more does he have?”*

8 Conclusion

We studied reasoning about quantities in natural language text. We have identified and defined an interesting and useful slice of the Textual Entailment problem, the Quantity Entailment task, and studied also quantitative reasoning problems that arise in elementary math word problems.

Our ability to support quantitative reasoning builds on a method we proposed for detecting and normalizing quantities in unrestricted English text; we developed a framework to remove variability and ambiguity from unstructured text by mapping it into a representation which makes reasoning more tractable. Once quantities are mapped into our representation we can support the reasoning required by Quantity Entailment and elementary school level math word problems. Our experiments exhibit quite impressive performance on a range of quantitative reasoning problems, including 87% success on solving math word problems that are targeted at elementary school kids.

Our future work will focus on alleviating some of the limitations of the inference module described in Section 5.2. We would also like to extend the scope of reasoning to the case of partially-ordered quantities, and focus on deeper semantic analysis to handle more involved math word problems.

Acknowledgments

This research was sponsored by the Army Research Laboratory (ARL) (under agreement W911NF-09-2-0053), DARPA (under agreement number FA8750-13-2-0008), and a grant from AI2. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the view of the agencies.

References

S. Banerjee, S. Chakrabarti, and G. Ramakrishnan. 2009. Learning to rank for quantity consensus queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 243–250, New York, NY, USA. ACM.

J. Barwise and R. Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*.

D. Bobrow. 1964. Natural language input for a computer problem solving system. Technical report, Cambridge, MA, USA.

A. Carlson, C. Cumby, J. Rosen, and D. Roth. 1999. The SNoW learning architecture. Technical report, UIUC Computer Science Department.

K.-W. Chang, R. Samdani, and D. Roth. 2013. A constrained latent variable model for coreference resolution. In *EMNLP*.

M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.

I. Dagan, O. Glickman, and B. Magnini, editors. 2006. *The PASCAL Recognising Textual Entailment Challenge.*

I. Dagan, D. Roth, M. Sammons, and F. Zanzotto. 2013. Recognizing textual entailment: Models and applications.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *ACL*.

Q. Do, W. Lu, and D. Roth. 2012. Joint inference for event timeline construction. In *EMNLP*.

K. Forbus. 1984. Qualitative process theory. *Artificial Intelligence*, 24:85–168.

Y. Freund and R. Schapire. 1998. Large margin classification using the Perceptron algorithm. In *COLT*.

K. Garoufi. 2007. Towards a better understanding of applied textual entailment: Annotation and evaluation of the rte-2 dataset. Master's thesis, Saarland University, Saarbrucken.

S. Kuehne. 2004a. On the representation of physical quantities in natural language text. In *Proceedings of Twenty-sixth Annual Meeting of the Cognitive Science Society*.

S. Kuehne. 2004b. *Understanding natural language descriptions of physical phenomena*. Ph.D. thesis, Northwestern University, Evanston, Illinois.

N. Kushman, L. Zettlemoyer, R. Barzilay, and Y. Artzi. 2014. Learning to automatically solve algebra word problems. In *ACL (1)*, pages 271–281.

I. Lev, B. Maccartney, C. Manning, and R. Levy. 2004. Solving logic puzzles: From robust processing to precise semantics. In *In Proc. of 2nd Workshop on Text Meaning and Interpretation, ACL-04*.

Bill Maccartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*.

R. Montague. 1973. The proper treatment of quantification in ordinary english. In Patrick Suppes, Julius Moravcsik, and Jaakko Hintikka, editors, *Approaches to Natural Language*, volume 49, pages 221–242. Dordrecht.

A. Mukherjee and U. Garain. 2008. A review of methods for automatic understanding of natural language mathematical problems. *Artificial Intelligence Review*, 29(2):93–122.

M. Palmer, D. Gildea, and N. Xue. 2010. *Semantic Role Labeling*.

I. Pratt-Hartmann. 2005. From timeml to TPL. In *Annotating, Extracting and Reasoning about Time and Events, 10.-15. April 2005*.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *NIPS*.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*.

W. Purdy. 1991. A logic for natural language. *Notre Dame Journal of Formal Logic*, 32(3):409–425, 06.

J. Pustejovsky, J. Castao, R. Ingria, R. Saur, R. Gaizauskas, A. Setzer, and G. Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *in Fifth International Workshop on Computational Semantics (IWCS-5*.

D. Roth and D. Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING-ACL, The 17th International Conference on Computational Linguistics*.

M. Sammons, V.G. Vydiswaran, and D. Roth. 2010. "ask not what textual entailment can do for you...". In *ACL*.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *NIPS*.

R. Saur, R. Knippen, M. Verhagen, and J. Pustejovsky. 2005. Evita: a robust event recognizer for qa systems. In *Proceedings of the conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pages 700–707, Stroudsburg, PA, USA. Association for Computational Linguistics.

U. Schwertel. 2003. *Plural Semantics for Natural Language UnderstandingA Computational Proof-Theoretic Approach*. Ph.D. thesis, University of Zurich.