Large-Scale Extraction and Use of Knowledge From Text

Peter Clark and Phil Harrison
Boeing Networked Systems Technology
The Boeing Company, PO Box 3707, Seattle, WA 98124
{peter.e.clark,philip.harrison}@boeing.com

ABSTRACT

Many AI tasks, in particular natural language processing, require a large amount of world knowledge to create expectations, assess plausibility, and guide disambiguation. However, acquiring this world knowledge remains a formidable challenge. Building on ideas by Schubert, we have developed a system called DART (Discovery and Aggregation of Relations in Text) that extracts simple, semi-formal statements of world knowledge (e.g., "airplanes can fly", "people can drive cars") from text by abstracting from a parser's output, and we have used it to create a database of 23 million propositions of this kind. An evaluation of the DART database on two language processing tasks (parsing and textual entailment) shows that it improves performance, and a human evaluation shows that over half the facts in it are considered true or partially true, rising to 70% for facts seen with high frequency. The significance of this work is two-fold: First it has created a new, publically available knowledge resource for language processing and other data interpretation tasks, and second it provides empirical evidence of the utility of this type of knowledge, going beyond Schubert et al's earlier evaluations which were based solely on human inspection of its contents.

Categories and Subject Descriptors

I.2.7 Natural Language Processing; I2.6 Learning – *knowledge acquisition*

General Terms

Algorithms, Experimentation

Keywords

Knowledge acquisition, parsing, textual entailment, natural language processing, commonsense knowledge, information extraction.

INTRODUCTION

A large amount of world knowledge is essential for many interpretation tasks, in particular natural language processing, to create expectations that can help assess plausibility and guide disambiguation. While a small number of repositories of (certain types of) world knowledge exist, e.g., Cyc [8], WordNet [5], ConceptNet [7], and FrameNet [2], we have followed work by Schubert et al., [17,21] who proposed that it might be possible to extract simple world knowledge propositions from text (e.g., "airplanes can fly", "houses can have roofs", "doors can be opened", "trees can shed leaves") from abstracted parse structures. This semi-formal, propositional knowledge complements other resources by providing a vast number of examples of "the way the world can be" (as revealed in text). These can then be used for many purposes, for example to assess plausiblity during disambiguation and information interpretation, to guide a system towards more plausible interpretations.

In this paper we present our work on generating a large database of "tuples" that encode this kind of general knowledge, using a system called DART (Discovery and Aggregation of Relations in Text), following similar techniques to Schubert et al. but expanding the types of tuples extracted and the volume of text used. To evaluate the resulting database, we have performed two case studies to see if it improves performance, namely for parsing and for textual entailment. For parsing, we use the database to bias a parser to prefer structures which have been seen often before as reflected in the database, a generalization of the way dependency triples were used to bias the MiniPar parser [11]. For recognizing entailment, we use the database to help compute the plausibility of inference rule instantiations, again by considering rule clauses that have been seen often before as more plausible. In both cases, the DART database produced small but statistically significant improvements, thus providing evidence that this type of world knowledge has practical value. In addition, a human evaluation of the database shows that over half the facts in it are considered true or partially true, rising to 70% for facts seen with high (> 10) frequency. The contribution of this work is thus a new, public domain knowledge resource that appears to be useful and of reasonable quality, and that expands on Schubert's original work in terms of the types of tuples extracted, the volume of data collected, and the evaluation of the results.

EXTRACTING KNOWLEDGE FROM TEXT

Schubert's Conjecture

Our work follows Schubert's conjecture that "there is a largely untapped source of general knowledge in texts, lying at the level beneath the explicit assertional content." [17]. This conjecture is based on the observation that a sentence such as:

"The camouflaged helicopter landed near the embassy"

not only describes a specific event, but also implicitly reveals more general knowledge, such as: helicopters can land, and helicopters can be camouflaged. By parsing sentences and abstracting from the resulting syntactic structure (e.g., dropping modifiers), large numbers of general propositions can be obtained. While some will be erroneous (e.g., due to misparses), statistically higher frequency propositions can be considered more reliable, and hence statistics used to assign a confidence value to each proposition. Schubert and Tong [18] followed this approach and showed that many propositions obtained this way were considered reasonable by a panel of human reviewers. Van Durme and Schubert subsequently created a proposition database using a system called KNEXT (KNowledge EXtraction from Text) [21]. While the approach has some overlap with work on constructing specific databases such as two-word collocations [10] and selectional preferences [16], it differs in that it collects a variety of larger structures, and makes (and assesses) claims about the resulting resource as a source of world knowledge in its own right. Banko et al. [3] have performed similar work in the TextRunner system, running over more text but using lighterweight linguistic analysis (part-of-speech tagging and noun phrase chunking rather than full parsing), producing less normalized and less structured data but at a faster rate. Our work follows Schubert et al's general knowledge extraction methodology, differing in some of the details and producing a large resource of 23 million distinct propositions.

Our database is created by a system called DART (Discovery and Aggregation of Relations in Text). DART parses text using SAPIR, a hand-built, broad coverage phrase structure parser [6], and then extracts "tuples" from the parse tree. Subsequently, tuples are factored into 12 different knowledge types, resulting in 12 types of proposition in the resulting database. We now describe these steps, then later describe an evaluation of the resulting database.

A tuple is a data structure containing a piece of syntactic information extracted from text, produced during parsing by a set of tuple generation rules attached to grammar rules. Syntactically, each tuple can be thought of as a fragment of the parse tree that retains head words or embedded tuples from a parse subtree, for example

"The camouflaged helicopter landed"

produces the tuples

(**AN** "camouflaged" "helicopter")
(**S** "helicopter" "land")

(where **AN** denotes an "adjective-noun" tuple and **S** denotes a sentential tuple). Semantically, each tuple denotes an example of a combination that exists in the world (as reflected by the parsed corpus), and thus can be thought of as a statement of possibility, e.g., "helicopters can be camouflaged" and "helicopters can land".

As a data structure, each tuple consists of a tuple type symbol, followed by arguments that typically are (root form) words from the input sentence, or, in some cases, other tuples. For example, the sentence "The lazy men from the city walked to the fancy store" produces four tuples:

Tuples for "The lazy men from the city walked to the fancy store"
(**S** "man" "walk" (**PP** "to" "store"))
(**NPN** "man" "from" "city")
(**AN** "lazy" "man")
(**AN** "fancy" "store")

which (following Schubert [17]) can be interpreted as: "men can walk to stores", "men can be from cities", "men can be lazy", and "stores can be fancy". DART generates six types of structure in all:

(**S** subj verb [obj] [(**PP** prep pobj)*]) ; sentential
(**AN** adj noun) ; adjective-noun
(**NPN** noun prep noun) ; noun-prep-noun
(**NN** noun noun) ; noun-noun
(**QN** (quantity unit) noun) ; quantity noun
(**S-ADJ** noun adj [(**PP** prep pobj)*]);sentential adj

where the arguments are words, or (**PN** *word*) for proper nouns, or NIL if absent, or an entire **S** structure for sentential complements. Root forms (e.g. "man") replace the original plural and inflected verb forms. **S** tuples contain the (head word of the) subject, verb, and, if present, the object, and **PP** tuples associated with prepositional phrases attached to the verb. **AN** tuples denote adjective-noun modification. **QN** denote quantified nouns, e.g., ((**QN** "10" "day") "visit") from "10 day visit". **S-ADJ** denote sentential adjectives, e.g., (**S-ADJ** "car" "red") from "The car is red". An example from real data is:

Tuples for "As Curran has forcefully shown, advertising also plays a part in shaping a newspaper."
(**S** (**PN** "Curran") "show")
(**S** "advertising" "play" "part"
(**PP** "in" (**S** NIL "shape" "newspaper")))
(**S** NIL "shape" "newspaper")

Note that DART does not capture all grammatical elements (not all grammar rules have tuple generation), for example adverbs are not currently captured in the tuples. Also note that **S** tuples that are derived from gerundive NPs (from -ing verb phrases) can appear in the object position of other tuple types.

In addition to simply extracting head words and dropping modifiers, DART will perform additional transformations, for example it restores head words of fronted wh-noun phrases, as illustrated below:

Tuples for "Which book did John bring?"
(S (**PN** "John") "bring" "book")

Coordination structures are handled by computing all tuple combinations over the head words from each coordinated phrase, for example:

Tuples for "The men and women ate beans and rice"
(**S** "man" "eat" "bean")
(**S** "woman" "eat" "bean")
(**S** "man" "eat" "rice")
(**S** "woman" "eat" "rice")

Tuple Proposition	Verbalization	Frequency
(**AN** "small" "hotel")	"Hotels can be small."	144
(**ANPN** "subject" "agreement" "to" "approval")	"Agreements can be subject to approvals."	121
(**NN** "drug" "distributor")	"There can be drug distributors."	17
(**NV** "bus" "carry")	"Buses can carry [something/someone]."	153
(**NPN** "sentence" "for" "offence")	"Sentences can be for offences."	26
(**NVN** "critic" "claim" "thing")	"Critics can claim things."	119
(**NVPN** "person" "go" "into" "room")	"People can go into rooms."	192
(**NVNPN** "democrat" "win" "seat" "in" "election")	"Democrats can win seats in elections."	11
(**QN** "year" "contract")	"Contracts can be measured in years."	1572
(**VN** "find" "spider")	"Spiders can be found."	8
(**VPN** "refer" "to" "business")	"Referring can be to businesses."	14
(**VNPN** "educate" "person" "at" "college")	"People can be educated at colleges."	103

Figure 1: Examples of the 12 types of propositions in the DART database.

Generating the DART Database

We ran DART over 2 text corpora, namely the Reuters corpus[1] and the British National Corpus (BNC)[2], producing 53 million raw tuples in about two CPU months. From these, we decomposed the more complex structures into 12 kinds of proposition, listed below and exemplified in Figure 1. Proper nouns are replaced with "person," "place," or "organization" using a named entity recognizer, pronouns are replaced with "person" or "thing" depending on the pronoun gender, and embedded propositions with "thing". The 12 types of proposition along with a template for verbalizing their meaning in English are as follows:

(**AN** *A N*)	"*N*s can be *A*."
(**ANPN** *A* N_1 *P* N_2)	"N_1s can be *A P* N_2s."
(**NN** N_1 N_2)	"There can be N_1 N_2."
(**NV** *N V*)	"*N*s can *V* [something/someone]."
(**NPN** N_1 *P* N_2)	"N_1s can be *P* N_2s."
(**NVN** N_1 *V* N_2)	"N_1s can *V* N_2s."
(**NVPN** N_1 *V P* N_2)	"N_1s can *V P* N_2s."
(**NVNPN** N_1 *V* N_2 *P* N_3)	"N_1s can *V* N_2s *P* N_3s."
(**QN** *UNIT N*)	"*N*s can be measured in *UNIT*s."
(**VN** *V N*)	"*N*s can be *V*ed."
(**VPN** *V P N*)	"*V*ing can be *P N*s."
(**VNPN** *V* N_1 *P* N_2)	"N_1s can be *V*ed *P* N_2s."

where *N,V,A*, and *P* denote a noun, verb, adjective, and preposition respectively, and the "s" suffix denotes pluralization. The final DART database contains 23 million unique propositions of these 12 kinds (110 million unaggregated). Examples are shown in Figure 1.

Using the Database

To measure the "strength" (plausibility) of a proposition (tuple), i.e., how much the observed frequency indicates a genuine association between the words rather than one that could have arisen by chance, we use a mutual information measure (following others, e.g., Alshawi and Carter [1]; Lin [10]). The mutual information *MI* of a tuple, e.g., (**NN** w_1 w_2), compares the actual and expected probabilities of the tuple if w_1 and w_2 were chosen at random:

$$MI = \log \frac{p(\mathbf{NN}\ w_1\ w_2)}{p(\mathbf{NN}\ w_1\ *).p(\mathbf{NN}\ *\ w_2)}$$

where p(**NN** w_1 w_2) is the probability of an **NN** tuple being (**NN** w_1 w_2), p(**NN** w_1 *) is the probability of an **NN** tuple having w_1 as its first argument, and p(**NN** * w_2) is the probability of an **NN** tuple having w_2 as its second argument. Following Lin [10], p(**NN** w_1 w_2) is measured from frequencies in the DART database using the formula:

$$p(\mathbf{NN}\ w_1\ w_2) = \frac{f(\mathbf{NN}\ w_1\ w_2) - c}{f(\mathbf{NN}\ *\ *)}$$

where f(**NN** w_1 w_2) is the frequency of (**NN** w_1 w_2) tuples in the database, f(**NN** * *) is the total number of **NN** tuples, and c is a constant (following Lin, we also use 0.95) to reduce problems of over-estimation for rare, observed events. We also constrain mutual information to be ≥ 0 (i.e., discount evidence of negatively associated words) to reduce problems of under-estimation for rare, unobserved events. p(**NN** w_1 *) and p(**NN** * w_2) are computed via:

$$p(\mathbf{NN}\ w_1\ *) = p(w_1)\ .\ f(\mathbf{NN}\ *\ *)$$
$$p(\mathrm{NN}\ *\ w_2) = p(w_2)\ .\ f(\mathbf{NN}\ *\ *)$$

where p(w_1) and p(w_2) (with their appropriate part of speech) come from a word frequency database[3]. Similar computations are used to compute mutual information for other types of tuples.

EVALUATION

While Schubert et al's earlier evaluations of tuple-style world knowledge relied on human inspection, our goal has been to go further and explore its utility in specific lan-

[1] http://trec.nist.gov/data/reuters/reuters.html

[2] http://www.natcorp.ox.ac.uk/

[3] http://www.kilgarriff.co.uk/bnc-readme.html

guage processing tasks. Note that our goal is to assess the DART database by seeing if it can improve performance in some example natural language processing (NLP) applications, rather than making specific claims about the example NLP applications themselves. Our conjecture is that tuple-style knowledge is useful for a broad range of applications, and the two examples we have evaluated provide support for that conjecture.

Improving parsing

Method

The first task we used to assess the database is for parsing. Previous work has demonstrated that specific types of collocation data can improve parsing. For example, Ratnaparkhi [15] acquires (verb,prep,noun) and (noun,prep,noun) tuples from unambiguous attachments found in a training corpus, then uses them to guide ambiguous attachments. Similarly, the MiniPar parser (as described in [11]) uses dependency triple (collocation) frequencies found from parsing a 1GB newswire corpus to guide future parsing. Our goal is thus to see if the database can similarly improve parsing, generalizing from this previous work to use the full range of tuple structures to provide statistical bias.

For this evaluation we again used the SAPIR parser, although the approach could be applied to any parser, whether hand-built or trained on treebanks. To exploit the tuples, a method must be devised to bias the parser's preferences with the statistical expectations from the database, and that method will vary depending on which parser is used. For SAPIR, we used the following method. Normally, SAPIR selects parses based on a set of cost functions attached to the grammar rules that return a small positive or negative number (negative is favorable, positive is unfavorable) for each node. The system selects the tree with the minimum total cost as the preferred parse. To take account of tuples, the cost functions were modified to assess the "cost" of the tuples associated with each node. That is, for nodes that could potentially be used to generate a tuple-like structure, the parser looks in the DART database to find how many such tuples there actually are, and adjusts the cost function accordingly based on the number found, reflecting that this part of the parse is "more likely" if it has been seen often before. There are many ways this adjustment could be made, and we used a simple approach as follows. In SAPIR, the values of the costing function typically vary between -2 ("moderately strong" preference) and 0 (no preference) (in addition, positive values are used to express negative preference). Similarly, the Mutual Information (MI) values of the tuples typically lie between 6 (high) and 0 (no information). We thus assign the "cost" of the tuples:

$$\text{Cost(tuple)} = -\text{MI(tuple)}/3$$

This provides a simple linear mapping between the MI values and the cost values.

Example

An example of how the DART database can improve parsing, the sentence:

> "Worn parts of old dresses were replaced with contrasting fabric and unfashionable outer and under garments were unpicked and remade in more fashionable styles."

was, without tuples, incorrectly parsed as the coordination of two sentences, the first ending with the word "outer", i.e., "(worn...outer) and (under...styles)". Also, the word "contrasting" was incorrectly analyzed as a verb. With tuples, the database contains support for (i.e., examples of) the tuple (**AN** "outer" "garment"), leading the parser to construct two sentences with the first correctly ending at the word "fabric", i.e., "(worn...fabric) and (unfashionable...styles)". Also, the tuple (**AN** "contrasting" "fabric") in the database caused "contrasting" to be correctly analyzed as an adjective.

Evaluation

To evaluate whether the DART database might improve parsing, we compared parses without and with the database on the Brown corpus (one million words) [12,13], which contains a "gold standard" parse for each sentence. We ignored sentences that had no SAPIR parse, as no comparisons without vs. with the database could be made for these sentences. To score a parse, we compare it with the gold standard parse in the Brown corpus as follows. First, the comparison program walks down each parse tree and at each node, collects a "parse fragment", namely the node label and the phrase represented by the subtree. For example, for the parse of the sentence below in which the prepositional phrase “with a fork” is (desirably) attached to the verb "ate", these fragments are:

"The man ate spaghetti with a fork"
(NP "The man")
(VP "ate spaghetti with a fork")
(VP "ate spaghetti")
(NP "spaghetti")
(PP "with a fork")
(NP "a fork")

whereas if the phrase had instead been attached to "spaghetti", we would see (NP "spaghetti with a fork") rather than (VP "ate spaghetti") in the list of fragments. Then, to compute a score for a parse we compare its fragments with the fragments from the gold standard parse, counting the number correct (SAPIR fragments also in Brown), incorrect (SAPIR fragments not in Brown), and missing (Brown fragments not in SAPIR). We define:

RelativePrecision = Correct/(Correct + Incorrect)
RelativeRecall = Correct/(Correct + Missing)

Note that as there are small differences in the underlying grammars, these are measures of relative precision and recall, i.e., even with a perfect SAPIR parse the scores may not be 1.0 (e.g., SAPIR may produce phrases with labels that are not used in the Brown corpus).

	Mean Relative Precision (%):	Mean Relative Recall (%):
Without tuples:	46.3	76.9
With tuples:	47.1	77.6

Table 1: The DART database produces a small but significant increase in relative precision and relative recall on the Brown corpus.

Results

The mean relative precision and recall are shown in Table 1. Although the improvements are small, they are statistically significant (>99.9%, using a paired, two-tailed T test), and thus shows that the database can produce improvements in parsing. It is interesting that in some cases recall dropped, and from further investigation the primary cause appeared to be that despite having 23 million tuples, there were sometimes cases where tuple frequencies were zero or near zero, and thus the parser was not always correctly biased. This suggests that it would be advantageous to have even more tuples, and/or methods to aggregate tuple frequencies together, as we discuss later. Despite this, the overall gain was positive with high significance, and illustrates that the database can provide value in parsing.

Recognizing Textual Entailment

Method

The second task we use to assess the DART database is in recognizing textual entailment (RTE), where we use tuples to help assess the plausibility of instantiated inference (paraphrase) rules. The RTE task can be described as: given a text T, decide whether a hypothesis text H can be "reasonably concluded" from T. For example, given:

T: The president visited Iraq.
H: The president traveled to Iraq.

it is reasonable to conclude H given T, and so we say T *entails* H. To recognize such entailments, many RTE systems, including our own [4], use inference rule ("paraphrase") databases to supply some of the required lexical and world knowledge, the most popular being DIRT [11]. DIRT contains approximately 12 million machine-discovered rules of the form

IF X *relation* Y **THEN** X *relation'* Y

along with an associated confidence value, where *relation* is a dependency path between X and Y. An example rule is:

IF X visits Y **THEN** X travels to Y

However, without restrictions on X and Y these rules are typically over-general, and can be instantiated in unlikely or nonsensical ways. For example, consider the DIRT rule

IF X shoots Y **THEN** X injures Y

While the rule applied to "Fred shot Sue" produces the plausible conclusion "Fred injured Sue", it produces unusual/nonsensical conclusions applied to "Fred shot the gun" (implies "Fred injured the gun"), as we do not normally think of guns as being injured. More recent work on RTE inference rules has sought to constrain inference rule application to avoid this problem, for example by creating preferences for the types of X and Y, called "inferential selectional preferences" (ISPs) [14], or by comparing the contexts (word environments) of X, Y, and the rule itself in sentences from which the rule was was learned, with the context in the sentence where it is being applied [19]. Loosely speaking, the goal of these methods is to disprefer "unusual" rule application, i.e., unusual with respect to the sentences where the rule was learned. However, while these methods show some improvements, there is considerable scope for further tightening the bounds of rule application due to the sometimes limited information in the learning contexts (e.g., the WordNet-based ISP for "**IF** X shoots Y **THEN** X injures Y" still includes artifact#n#1 as one of Y's preferred types, thus allowing "Fred injured the gun"). Our goal here is to see if expectations from the DART database can also contribute to identifying "good" instantiations. For example, the DART database includes the expectation "people can be injured" (1847 examples), but no examples of "guns can be injured".

To assess the DART database's utility for this task, we performed an experiment using our own RTE system [4] which uses the DIRT inference rules. Note that as with parsing, our goal here is to evaluate the DART database, not DIRT or the particular RTE system we are using. To use DART to assess a rule's instantiation, we measure how well the instantiation matches expectations in the database. Specifically, a rule's condition and action are each a dependency path between two variables, X and Y. After a rule is instantiated, each link in the instantiated path can be expressed as a tuple, e.g., the path in the instantiated condition "Fred shot the gun" can be expressed as 2 tuples (**NV** "person" "shoot") and (**VN** "shoot" "gun"). Similarly the instantiated action "Fred injured the gun" can be expressed as (**NV** "person" "injure") and (**VN** "injure" "gun"). Counts for these 4 tuples in the DART database are 28, 12, 31, and 0 respectively, producing Mutual Information (MI) values of 3.37, 2.56, 3.46, and 0.00. We take the overall plausibility of the rule as the minimum of the MI values (here 0.00), i.e., the rule is only as plausible as its least plausible part (In the case of a tie, we use the MI of the second least plausible tuple as a second-order ranking). Note that this approach only assesses the plausibility of (independently) the condition and action of a particular instantiation of a rule, not the implication itself nor the rule in general.

Computing Entailment

To recognize entailment, our software first parses the T and H sentences and generates a shallow logical representation (set of ground clauses) for each. It then computes whether (the logic for) T, or some DIRT-based implication of T, is subsumed by (the logic for) H, with up to one mismatching predicate allowed For example T:"A black cat devoured a mouse" is subsumed by (i.e., implies) H: "An animal ate", as (using WordNet's hypernym tree) a cat is an animal, and devouring is an eating event. Similarly T:"A black cat devoured a mouse" implies H: "The cat swallowed the mouse" via the DIRT rule "**IF** X eats Y **THEN** X swallows Y". Currently, the base system does not assign confidences to its conclusions. The overall system scores on the recent RTE tests (non-ranked 2-way) were 54.1% for RTE3[4] and 56.5% for RTE4[5].

Evaluation

We reran our RTE system on the RTE3 and RTE4 test sets, and for pairs where a DIRT rule was used to conclude entailment (104 pairs in RTE3, 138 pairs in RTE4) we compared three methods for assigning confidence to those conclusions. (We did not use other pairs as the base system itself does not assign confidences). These methods were:

a. randomly
b. the original DIRT rule confidence
c. the DART-derived instantiated rule confidence.

Given the confidences, we can produce a rank-ordering of the rule-based entailments. Ideally, all the positive entailments (as assigned by human judges) will be at the top of this list, followed by all the negative entailments. To assess the overall quality of a particular rank-ordering, a commonly used measure is "average precision" (AP) [22] defined as "the average of the system's precision values at all points in the ranked list in which recall increases, that is at all points in the ranked list for which the gold standard annotation is YES". More formally, it can be written:

$$\mathrm{AP} = \tfrac{1}{R}\sum_{i=1}^{n} E(i).\#correct_i\,/\,i$$

where n is the number of the pairs in the test set, $\#\mathrm{correct}_i$ is the number of positive pairs in the top i predictions, R is the total number of positive pairs in the test set, E(i) is 1 if the i-th pair is positive and 0 otherwise, and i ranges over the pairs, ordered by their ranking. In the perfect case, the top R cases will be the R positive pairs in the test set. The Average Precision results we obtained are shown in Table 2 (random was averaged over 100 runs).

Ranking Method:	AP of rule-based entailments for:	
	RTE3	RTE4
Random:	0.592	0.650
Original rule confidence:	0.634	0.712
Tuple (DART) derived confidence:	0.641	0.728

Table 2: Average Precision (AP) of ranked entailments when a DIRT rule was used.

As shown in Table 2, for both the RTE3 and RTE4 data sets the DART database was able to provide meaningful information about the reliability of instantiated rules compared with the base (random) case, again illustrating its ability to supply useful expectations. The level of information is roughly similar to using the DIRT-supplied confidences. For example, for the RTE4 pair #997:

> **T:** Michelin used the Avant as the test bed for their radial tyres.
>
> **H*:** Michelin invented radial tyres. [*NOT* entailed]

the DIRT rule

> **IF** X uses something as a Y **THEN** X attacks with Y

undesirably fired (thus concluding "Michelin attacks with a bed" (!)) which (with other knowledge) led it to incorrectly conclude H. However, the DART database rates this rule instantiation as very unlikely because there are no examples of (**VPN** "attack" "with" "bed"), and thus assigns minimal confidence to that conclusion. Again, this experiment illustrates that the database contains meaningful world knowledge which can play a role in applications such as this, potentially augmenting other approaches.

Human Assessment of the Database

As a final evaluation of the quality of the database, we divided the DART propositions into 4 buckets depending on their frequency in the database, namely: 1, 2-10, 11-100, and >100. 12 independent human judges then rated the quality of approximately 300 (different) propositions each, drawn randomly but in equal quantities from the four buckets, i.e., approximately 75 propositions per bucket per judge, for a total of 3634 propositions. Judges were asked to score propositions on a 1-5 scale:

1: true
2: partially true
3: unsure/unclear
4: dubious/mainly false
5: false

Judges were told to score incomprehensible or meaningless or misspelt (e.g., "People can be brillient.") propositions as as false (5), but not penalize poor pluralization (e.g., "bookses") as this is due to the verbalizer not the original proposition. Our interest is in what fraction of the DART database is subjectively considered "good" (i.e., contains true (1) or partially true (2) propositions). The results are shown in Figures 2 and 3, both for all 12 proposition types combined and individually. The results in Figure 2 show that more than 60% of the propositions seen two or more times were judged partially true/true, rising to approximately 70% for propositions seen at least 10 times. Similar trends are seen in Figure 3 for the individual proposition types. For just the true (1) category, the curve is very similar to Figure 2 but slightly (5%) lower. Although there is some subjectivity in the judging, the results are encouraging as they suggest a significant portion of the database is considered coherent.

[4] http://pascallin.ecs.soton.ac.uk/Challenges/RTE3
[5] http://www.nist.gov/tac/tracks/2008/rte/

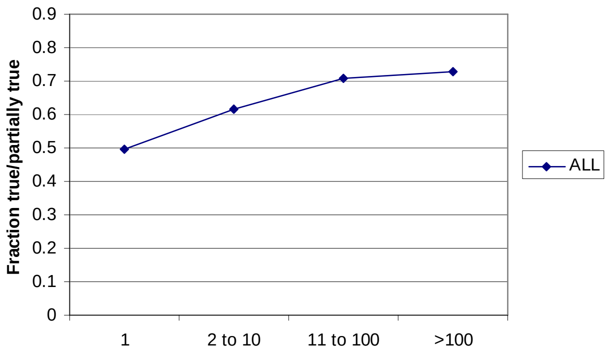

Figure 2: Frequency of proposition in DART database

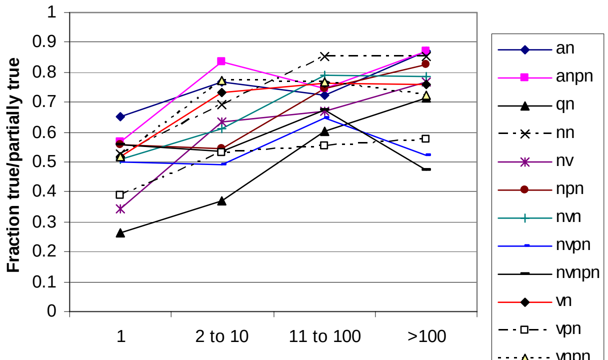

Figure 3: Frequency of proposition in DART database

DISCUSSION AND CONCLUSIONS

While there is new interest in creating general knowledge resources from text, there are still few such resources available. DART has produced a new, publically available resource containing a vast number of semi-formal, general propositions about the world, generated using a novel variant on Schubert's technique and extending the range of proposition types contained. Our evaluation shows that the database can be usefully employed for two language processing tasks, and is considered to contain a large proportion of true or partially true propositions by independent judges.

Of the few other existing knowledge resources, Schubert et al's KNEXT database [21], TextRunner [3] and ConceptNet [7,11] are the most similar to DART. As already discussed, our work here was inspired by KNEXT, and has extended that work with respect to the size, variety of tuples, application, and evaluation of that approach. Similarly mentioned earlier, TextRunner performs automatic extraction from text, but only outputs triples (essentially subject-verb-object phrases), and in a less normalized form as the triple arguments are typically unprocessed natural language phrases rather than words, for example:

("iodine" "will kill" "the lactic bacteria")
("my friends" "had" "a car")

This less structured form places a greater post-processing requirement on systems that wish to exploit its contents. Finally, ConceptNet [7] is a human authored (by Web volunteeers) resource, containing semi-formal propositions such as:

(CapableOf "a car" "crash")
(UsedFor "a car" "getting to work")

which can be verbalized, respectively, as "An activity a car can do is crash," and "A car is for getting to work." The primary differences with DART are that ConceptNet uses a fixed set of about 20 binary, semantic predicates (CapableOf, UsedFor, etc.), has less structured predicate arguments (like TextRunner, arguments can be arbitrary English phrases, not just words), is manually authored, and is two orders of magnitude smaller (~1 million facts, compared with ~23 million unique/~110 million with duplicates in DART). Although the structure and content of ConceptNet is different, it would be interesting to compare the reliability of facts in DART and ConceptNet, and also explore how the two resources could leverage each other (e.g., using one resource to help validate propositions in the other).

While there are numerous possible future directions, one of the most important appears to be finding ways to aggregate tuples together through generalization to increase the coverage of the database. Even with 23 million tuples, there are often cases where the tuple frequency (of "good" propositions) is zero or near zero, limiting DART's ability to provide guidance in such cases. However, there are often "similar" propositions in the database that could be leveraged. For example, although DART does not know that "giraffes can be in zoos", i.e., the frequency of the (**NPN** "giraffe" "in" "zoo") tuple is zero, it *does* know that lions, elephants, and monkeys can be in zoos, suggesting the generalization "animals can be in zoos", i.e., (**NPN** "animal" "in" "zoo"). Such generalizations would not only provide indirect evidence for zero-frequency propositions, but would also help sense-disambiguate the words themselves in the tuples. For example, given “lion”, “elephant” and “monkey” instantiate the same tuple pattern (**NPN** ?X “in” “zoo”), and senses lion#n1, elephant#n1, and monkey#n1 have a common supersense animal#n1 in WordNet, then “lion” most likely denotes the animal sense lion#n1 etc. Van Durme et al. [20] have conducted some exciting research in this direction that could also be applied to DART.

Similarly one part of the DART database could be used to disambiguate other parts. For example, although the tuple (**NN** "car" "engine"), verbalized as "There can be car engines", does not specify the nature of the noun-noun relation, other parts of DART do provide this, e.g., "Cars can be powered by engines", "Engines can be in cars". In general, there are large numbers of inter-proposition relationships within DART that could be discovered and harnessed to enhance the database.

Finally, an additional use for natural language understanding would be to use DIRT to help fill in implicit knowledge during text interpretation. For example, consider the sentence:

"John had studied hard, but he was still worried."

What might John be worried about? A query to the database (**ANPN** "worried" "person" "about" ?X) produces some possibilities:

?X = survival, selloff, plan, export, exam, delay, environment, career, ...

of which some (e.g., exam) may be correlated with other parts of the text (e.g., "studied") and thus be considered more likely. In this case, (**VPN** "study" "for" ?X) also returns ?X = exam, ..., thus increasing the likelihood that exams are the target of both the studying and worrying. In general, text gives partial knowledge of a scene, and for human readers our general knowledge can fill in the gaps. A database like DART can similarly help in this task, the multi-argument propositions playing a particularly useful role. We are thus optimistic that this resource can be a catalyst for further research in these and other exciting possible directions, and contribute to the wider goal of language understanding.

AVAILABILITY

The DART database is available at http://www.cs.utexas.edu/users/pclark/dart/

REFERENCES

[1] Alshawi, H., Carter, D. 1994. Training and Scaling Preference Functions for Disambiguation. *Computational Linguistics* 20 (4) pp635-648.

[2] Baker, C., Fillmore, C., and Lowe, J. 1998. "The Berkeley FrameNet Project." in Proc 36th ACL, pp86-90. CA:Kaufmann.

[3] Banko M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O. 2007. Open Information Extraction from the Web. *IJCAI'07.*

[4] P. Clark, P. Harrison. Recognizing Textual Entailment with Logical Inference. In *Proceedings of 2008 Text Analysis Conference (TAC'08)*, Gaithsburg, Maryland, 2008.

[5] Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

[6] Harrison, P., and Maxwell, M. 1986. A New Implementation of GPSG, *Proc. 6th Canadian Conf on AI (CSCSI'86)*, pp78-83.

[7] Havasi, C., Speer, R. & Alonso, J. 2007. ConceptNet3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. *Proceedings of Recent Advances in Natural Languges Processing*.

[8] Lenat, D. B., and Guha, R. V. 1990. *Building Large Knowledg Based Systems: Representation and Inference in the Cyc Project*. Reading, MA: Addison-Wesley.

[9] Lieberman, H., Liu, H., Singh. P., Barry, B. 2004. Beating some common sense into interactive applications. *AI Magazine*.

[10] Lin, D. 1998. Extracting Collocations from Text Corpora. *Workshop on Computational Terminology*. pp. 57-63.

[11] Lin, D., and Pantel, P. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering* 7 (4) pp 343-360.

[12] Marcus, M., Santorini, B., Marcinkiewicz, M. 1993. Building a Large Annotated Corpus of English : The Penn Treebank. *Computational Linguistics,* 19 (2). 313-330.

[13] Nelson, F., Kucera, H. 1982. Frequency analysis of English usage. Houghton Mifflin Company, Boston.

[14] Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., Hovy, E. 2007. ISP: Learning Inferential Selectional Preferences. In Human Language Technologies, NAACL HLT 2007.

[15] Ratnaparkhi, A. 1998. Unsupervised Statistical Models for Prepositional Phrase Attachment. *Proc. COLING-ACL'98*

[16] Resnik, P. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?,* pages 52-57.

[17] Schubert, L. 2002. Can we derive general world knowledge from texts?", M. Marcus (ed.), *Proc. of the 2nd Int. Conf. on Human Language Technology Research (HLT 2002),*

[18] Schubert, L. and Tong, M. 2003. Extracting and evaluating general world knowledge from the Brown corpus, *Proc. of the HLT/NAACL 2003 Workshop on Text Meaning.*

[19] Szpektor, I., Dagan, I., Bar-Haim, R., Goldberger, J. 2008. Contextual Preferences. Proceedings of ACL 2008.

[20] Van Durme, B., Michalak, P., Schubert, L. 2009. Deriving Generalized Knowledge from Corpora using WordNet Abstraction. *Proc. EACL'09.*

[21] Van Durme, B., Schubert, L. Open Knowledge Extraction through Compositional Language Processing. Symposium on Semantics in Systems for Text Processing (STEP'08). Venice, Italy. September 22-24, 2008.

[22] Voorhees E., and Harman, D. 1999. Overview of the seventh text retrieval conference. In *Proceedings of the Seventh Text Retrieval Conference (TREC-7).* NIST Special Publication.