Philosophers are Mortal: Inferring the Truth of Unseen Facts

Gabor Angeli
Stanford University
Stanford, CA 94305
angeli@stanford.edu

Christopher D. Manning
Stanford University
Stanford, CA 94305
manning@stanford.edu

Abstract

Large databases of facts are prevalent in many applications. Such databases are accurate, but as they broaden their scope they become increasingly incomplete. In contrast to extending such a database, we present a system to query whether it contains an arbitrary fact. This work can be thought of as re-casting open domain information extraction: rather than growing a database of known facts, we smooth this data into a database in which *any* possible fact has membership with some confidence. We evaluate our system predicting held out facts, achieving 74.2% accuracy and outperforming multiple baselines. We also evaluate the system as a common-sense filter for the ReVerb Open IE system, and as a method for answer validation in a Question Answering task.

1 Introduction

Databases of facts, such as Freebase (Bollacker et al., 2008) or Open Information Extraction (Open IE) extractions, are useful for a range of NLP applications from semantic parsing to information extraction. However, as the domain of a database grows, it becomes increasingly impractical to collect completely, and increasingly unlikely that all the elements intended for the database are explicitly mentioned in the source corpus. In particular, common-sense facts are rarely explicitly mentioned, despite their abundance. It would be useful to infer the truth of such unseen facts rather than assuming them to be implicitly false.

A growing body of work has focused on automatically extending large databases with a finite set of additional facts. In contrast, we propose a system to generate the (possibly infinite) completion of such a database, with a degree of confidence for each unseen fact. This task can be cast as querying whether an arbitrary element is a member of the database, with an informative degree of confidence. Since often the facts in these databases are devoid of context, we refine our notion of *truth* to reflect whether we would assume a fact to be true without evidence to the contrary. In this vein, we can further refine our task as determining whether an arbitrary fact is *plausible* – true in the absence contradictory evidence.

In addition to general applications of such large databases, our approach can further be integrated into systems which can make use of probabilistic membership. For example, certain machine translation errors could be fixed by determining that the target translation expresses an implausible fact. Similarly, the system can be used as a soft feature for semantic compatibility in coreference; e.g., the types of phenomena expressed in Hobbs' selectional constraints (Hobbs, 1978). Lastly, it is useful as a common-sense filter; we evaluate the system in this role by filtering implausible facts from Open IE extractions, and filtering incorrect responses for a question answering system.

Our approach generalizes word similarity metrics to a notion of *fact similarity*, and judges the membership of an unseen fact based on the aggregate similarity between it and existing members of the database. For instance, if we have not seen the fact that philosophers are mortal[1] but we know that Greeks are mortal, and that philosophers and Greeks are similar, we would like to infer that the fact is nonetheless plausible.

We implement our approach on both a large open-domain database of facts extracted from the Open IE system ReVerb (Fader et al., 2011), and ConceptNet (Liu and Singh, 2004), a hand curated database of common sense facts.

[1] This is an unseen fact in http://openie.cs.washington.edu.

2 Related Work

Many NLP applications make use of a knowledge base of facts. These include semantic parsing (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Kate et al., 2005; Zettlemoyer and Collins, 2007) question answering (Voorhees, 2001), information extraction (Hoffmann et al., 2011; Surdeanu et al., 2012), and recognizing textual entailment (Schoenmackers et al., 2010; Berant et al., 2011).

A large body of work has been devoted to creating such knowledge bases. In particular, Open IE systems such as TextRunner (Yates et al., 2007), ReVerb (Fader et al., 2011), Ollie (Mausam et al., 2012), and NELL (Carlson et al., 2010) have tackled the task of compiling an open-domain knowledge base. Similarly, the MIT Media Lab's ConceptNet project (Liu and Singh, 2004) has been working on creating a large database of common sense facts.

There have been a number of systems aimed at automatically extending these databases. That is, given an existing database, they propose new relations to be added. Snow et al. (2006) present an approach to enriching the WordNet taxonomy; Tandon et al. (2011) extend ConceptNet with new facts; Soderland et al. (2010) use ReVerb extractions to enrich a domain-specific ontology. We differ from these approaches in that we aim to provide an exhaustive completion of the database; we would like to respond to a query with either membership or lack of membership, rather than extending the set of elements which are members.

Yao et al. (2012) and Riedel et al. (2013) present a similar task of predicting novel relations between Freebase entities by appealing to a large collection of Open IE extractions. Our work focuses on arguments which are not necessarily named entities, at the expense of leveraging less entity-specific information.

Work in classical artificial intelligence has tackled the related task of loosening the closed world assumption and monotonicity of logical reasoning, allowing for modeling of unseen propositions. Reiter (1980) presents an approach to leveraging default propositions in the absence of contradictory evidence; McCarthy (1980) defines a means of overriding the truth of a proposition in abnormal cases. Perhaps most similar to this work is Pearl (1989), who proposes approaching non-monotonicity in a probabilistic framework, and in particular presents a framework for making inferences which are not strictly entailed but can be reasonably assumed. Unlike these works, our approach places a greater emphasis on working with large corpora of open-domain predicates.

3 Approach

At a high level, we are provided with a large database of facts which we believe to be true, and a query fact not in the database. The task is to output a judgment on whether the fact is plausible (true unless we have reason to believe otherwise), with an associated confidence. Although our approach is robust to unary relations, we evaluate only against binary relations.

We decompose this decision into three parts, as illustrated in Figure 1: (i) we find candidate facts that are similar to our query, (ii) we define a notion of similarity between these facts and our query, and (iii) we define a method for aggregating a collection of these similarity values into a single judgment. The first of these parts can be viewed as an information retrieval component. The second part can be viewed as an extension of word similarity to fact similarity. The third part is cast as a classification task, where the input is a set of similar facts, and the decision is the confidence of the query being plausible.

We define a fact as a triple of two arguments and a relation. We denote a fact in our database as $f = (a_1, r, a_2)$. A fact which we are querying is denoted by f_q – as our focus is on unseen facts, this query is generally not in the database.

3.1 Finding Candidate Facts

Naïvely, when determining the correctness of a query fact, it would be optimal to compare it to the entire database of known facts. However, this approach poses significant problems:

1. The computational cost becomes unreasonable with a large database, and only a small portion of the database is likely to be relevant.
2. The more candidates we consider the more opportunities we create for false positives in finding similar facts. For a sufficiently large database, even a small false positive rate could hurt performance.

To address these two problems, we consider only facts which match the query fact in two of their three terms. Formally, we define

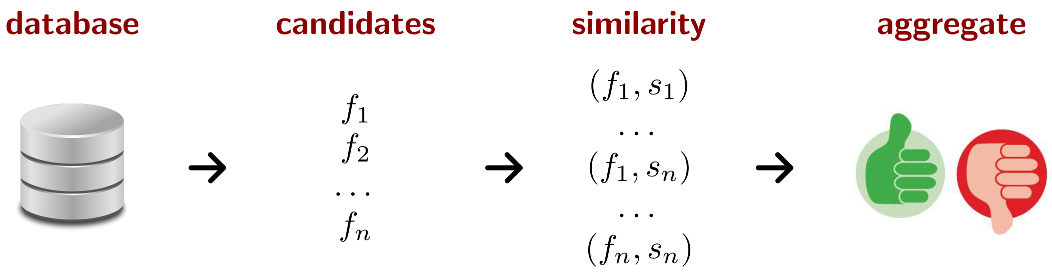

Figure 1: An overview of our approach. A large database of facts is queried for candidate entries that may be similar to the query fact (see Section 3.1); the similarity of each of these facts to the query fact is computed using a number of word similarity metrics (see Section 3.2); finally, these similarity judgments are aggregated into a single judgment per metric, and then a single overall judgment (see Section 3.3).

functions: $\text{cand}(f_q, f_i; a_1)$, $\text{cand}(f_q, f_i; r)$, and $\text{cand}(f_q, f_i; a_2)$ for whether the query f_q matches a fact in our database f_i on all but one of the arguments (or relation). For efficiency, the total number of candidates returned by each of these three functions was limited to 100, creating up to 300 similar facts overall.

The simplest implementation of this cand function would be exact match ($\text{cand}_{\text{exact}}$); however, this is liable to return few results. As an example, suppose our query is (*private land*, *be sold to*, *government*). We would like to consider a fact in our database (***his** land*, *be sold to*, *United States*) as similar except for second argument (*government* versus *United States*), despite the first argument not matching exactly. To account for this, we define a class of functions which match the head word of the two phrases, and as many of the following stricter criteria as possible while maintaining at least 40 candidate facts:[2]

$\text{cand}_{\text{head}}$ Match the head word of the two phrases only. Head words were extracted using the Stanford Parser (Klein and Manning, 2003), treating each argument and relation as a sentence.

cand_{vn} Match all verbs and nouns in the two phrases; This prunes candidates such as (*land **of our ancestors***, *be sold to*, *Prussia*). Tagging was done with the Stanford Tagger (Toutanova et al., 2003).

$\text{cand}_{\text{open}}$ Match the open-class words between the two phrases. More precisely, it matches every word which is not a pronoun, determiner, preposition, or form of the verb *be*. This prunes candidates such as (***worthless** land*, *be sold to*, *gullible investors*).

We proceed to describe our notion of similarity between facts, which will be applied to the set of candidate similar facts retrieved.

3.2 Similarity Between Facts

Determining the similarity between two facts is in general difficult. For sufficiently complicated facts, it can be has hard as recognizing textual entailment (RTE); for instance, determining that *every philosopher is mortal* and *Socrates is mortal* are similar requires fairly sophisticated inference. We choose a simple approach, in order to avoid fitting to a particular corpus or weakening our ability to generalize to arbitrary phrases.

Our approach casts fact similarity in terms of assessing word similarity. The candidate facts from Section 3.1 differ from the query fact by a single phrase; we define the similarity between the candidate and query fact to be the similarity between the differing term.

The word similarity metrics are summarized in Table 1. They fall into two broad classes: information-theoretic thesaurus based metrics, and distributional similarity metrics.

Thesaurus Based Metrics We adopt many of the thesaurus based similarity metrics described in Budanitsky and Hirst (2006). For each metric, we use the WordNet ontology (Miller, 1995) combined with n-gram counts retrieved from Google n-grams (Brants and Franz, 2006). Every word form was assigned a minimum count of 1; 2265 entries had no counts and were assigned this minimum (1.5%). 167 of these were longer than 5 words; the remaining did not appear in the corpus.

Since WordNet is a relatively sparse resource,

[2] This threshold is chosen in conjunction with the aggregation threshold in Section 3.3, to allow for at least two facts in the 95% threshold.

if a query phrase is not found a number of simple variants are also tried. These are, in order of preference: a lemmatized version of the phrase, the head word of the phrase, and the head lemma of the phrase. If none of these are found, then the named entities in the sentence were replaced with their types. If that fails as well, acronyms[3] were expanded. For words with multiple sense, the maximum similarity for any pair of word senses was used.

Distributional Similarity Based Metrics We define a number of similarity metrics on the 50 dimensional word vectors of Huang et al. (2012). These cover a vocabulary of 100,231 words; a special vector is defined for unknown words.

Compound phrases are queried by treating the phrase as a bag of words and averaging the word vectors of each word in the phrase, pruning out unknown words. If the phrase contains no known words, the same relaxation steps are tried as the thesaurus based metrics.

3.3 Aggregating Similarity

At this stage, we are presented with a set of candidate facts which may be similar to our query, and a set of similarity judgments for each of these candidate facts. Intuitively, we would like to mark a fact as plausible if it has enough sufficiently similar candidate facts based on a large number of metrics. This is a two-dimensional aggregation task: (i) we aggregate judgments for a single similarity metric, and (ii) we aggregate these aggregate judgments across similarity metrics. We accomplish the first half with a thresholded average similarity; the second half we accomplish by using the aggregate similarity judgments as features for a logistic regression model.

Thresholded Average Similarity Given a set of similarity values, we average the top 5% of the values and use this as the aggregate similarity judgment. This approach incorporates the benefit of two simpler aggregation techniques: averaging and taking the maximum similarity.

Averaging similarity values has the advantage of robustness – given a set of candidate facts, we would like as many of those facts to be as similar to the query as possible. To illustrate, we should be more certain that (*philosophers*, *are*, *mortal*) if we know both that (*Greeks*, *are*, *mortal*) and (*men*, *are*, *mortal*). However, since the number of similar facts is likely to be small relative the number of candidate facts considered, this approach has the risk of losing the signal in the noise of uninformative candidates. Taking the maximum similarity judgment alleviates this concern, but constrains the use of only one element in our aggregate judgment.

	Name	Formula
Thesaurus Based	Path	$-\log \text{len}(w_1, \text{lcs}, w_2)$
	Resnik	$-\log P(\text{lcs})$
	Lin	$\frac{\log(P(\text{lcs})^2)}{\log(P(w_1) \cdot P(w_2))}$
	Jiang-Conrath	$\log\left(\frac{P(\text{lcs})^2}{P(w_1) \cdot P(w_2)}\right)^{-1}$
	Wu-Palmer	$\frac{2 \cdot \text{depth}(\text{lcs})}{2 \cdot \text{depth}(\text{lcs}) + \text{len}(w_1, \text{lcs}, w_2)}$
Distributional	Cosine	$\frac{w_1 \cdot w_2}{\lVert w_1 \rVert \lVert w_2 \rVert}$
	Angle	$\arccos\left(\frac{w_1 \cdot w_2}{\lVert w_1 \rVert \lVert w_2 \rVert)}\right)$
	Jensen-Shannon	$\frac{(KL(p_1 \Vert p_2) + KL(p_2 \Vert p_1))}{2}$
	Hellinger	$\frac{1}{\sqrt{2}} \lVert \sqrt{p_1} - \sqrt{p_2} \rVert$
	Jaccard	$\frac{\lVert \min(w_1, w_2) \rVert_1}{\lVert \max(w_1, w_2) \rVert_1}$
	Dice	$\frac{\lVert \min(w_1, w_2) \rVert_1}{\frac{1}{2} \lVert w_1 + w_2 \rVert_1}$

Table 1: A summary of similarity metrics used to calculate fact similarity. For the thesaurus based metrics, the two synsets being compared are denoted by w_1 and w_2; the lowest common subsumer is denoted as lcs. For distributional similarity metrics, the two word vectors are denoted by w_1 and w_2. For metrics which require a probability distribution, we pass the vectors through a sigmoid to obtain $p_i = \frac{1}{1+e^{-w_i}}$.

If fewer than 20 candidates are returned, our combination approach reduces to taking the maximum similarity value. Note also that the 40 fact threshold in the candidate selection phase is chosen to provide at least two similarity values to be averaged together. The threshold was chosen empirically, although varying it does not have a significant effect on performance.

Aggregate Similarity Values At this point, we have a number of distinct notions of similarity: for each metric, for each differing term, we have a judgment for whether the query fact is similar to the list of candidates. We combine these using

[3] 6053 acronyms and initialisms were scraped from `http://en.wikipedia.org/wiki/List_of_acronyms_and_initialisms`

a simple logistic regression model, treating each judgment over different metrics and terms as a feature with weight given by the judgment. For example, cosine similarity may judge candidate facts differing on their first argument to have a similarity of 0.2. As a result, a feature would be created with weight 0.2 for the pair (cosine, argument 1). In addition, features are created which are agnostic to which term differs (e.g., the cosine similarity on whichever term differs), bringing the total feature count to 44 for 11 similarity metrics.

Lastly, we define 3 auxiliary feature classes:

- **Argument Similarity**: We define a feature for the similarity between the two arguments in the query fact. Similarity metrics (particularly distributional similarity metrics) often capture a notion more akin to relatedness than similarity (Budanitsky and Hirst, 2006); the subject and object of a relation are, in many cases, related in this sense.
- **Bias**: A single bias feature is included to account for similarity metrics which do not center on zero.
- **No Support Bias**: A feature is included for examples which have no candidate facts in the knowledge base.

4 Data

Our approach is implemented using two datasets. The first, described in Section 4.1, is built using facts retrieved from running the University of Washington's ReVerb system run over web text. To showcase the system within a cleaner environment, we also build a knowledge base from the MIT Media Lab's ConceptNet.

4.1 ReVerb

We created a knowledge base of facts by running ReVerb over ClueWeb09 (Callan et al., 2009). Extractions rated with a confidence under 0.5 were discarded; the first billion undiscarded extractions were used in the final knowledge base. This resulted in approximately 500 million unique facts.

Some examples of facts extracted with ReVerb are given in Table 2. Note that our notion of plausibility is far more unclear than in the ConceptNet data; many facts extracted from the internet are explicitly false, and others are true only in specific contexts, or are otherwise underspecified.

Argument 1	Relation	Argument 2
cat	Desires	tuna fish
air	CapableOf	move through tiny hole
sneeze	HasA	allergy
person who	IsA	not wage-slaves get more sleep

Table 3: Example ConceptNet extractions. The top rows correspond to characteristic correct extractions; the bottom rows characterize the types of noise in the data.

4.2 ConceptNet

We also created a dataset using a subset of ConceptNet. ConceptNet is a hand-curated common sense database, taking information from multiple sources (including ReVerb) and consolidating them in a consistent format. We focus on the manually created portion of the database, extracted from sources such as the Open Mind Common Sense[4] (Singh et al., 2002).

The knowledge base consists of 597,775 facts, each expressing one of 34 relations. Examples of facts in the ConceptNet database are given in Table 3. While the arguments are generally cleaner than the ReVerb corpus, there are nonetheless instances of fairly complex facts.

4.3 Training Data

Our training data consists of a set of tuples, each consisting of a fact f and a database d which does not contain f. We create artificial negative training instances in order to leverage the standard classification framework. We would like negative examples which are likely to be implausible, but which are close enough to known facts that we can learn a reasonable boundary for discriminating between the two. To this end, we sample negative instances by modifying a single argument (or the relation) of a corresponding positive training instance. In more detail: we take a positive training instance (a_1, r, a_2) and a fact from our database (a'_1, r', a'_2), and compute the cosine similarity $\text{sim}_{\text{cos}}(a_1, a'_1)$, $\text{sim}_{\text{cos}}(r, r')$, and $\text{sim}_{\text{cos}}(a_2, a'_2)$. Our negative instance will be one of (a'_1, r, a_2), (a_1, r', a_2), or (a_1, r, a'_2) corresponding to the entry whose similarity was the largest. Negative facts which happen to be in the database are ignored.

[4] http://openmind.media.mit.edu/

Argument 1	**Relation**	**Argument 2**
officials	contacted	students
food riots	have recently taken place in	many countries
turn	left on	Front Street
animals	have not been performed to evaluate	the carcinogenic potential of adenosine

Table 2: Example ReVerb extractions. The top rows correspond to characteristic correct extractions; the bottom rows shows examples of the types of noise in the data. Note that in general, both the arguments and the predicate can be largely unconstrained text.

To simulate unseen facts, we construct training instances by predicting the plausibility of a fact held out from the database. That is, if our database consists of $d = \{f_0, f_1, \dots f_n\}$ we construct training instances $(f_i, d \backslash \{f_i\})$. Negative examples are likewise constrained to not occur in the database, as are the facts used in their construction.

5 Results

We evaluate our system with three experiments. The first, described in Section 5.2, evaluates the system's ability to discriminate plausible facts from sampled implausible facts, mirroring the training regime. The second evaluates the system as a semantic filter for ReVerb extractions, tested against human evaluations. The third uses our system for validating question answering responses.

5.1 Baselines

We define a number of baselines to compare against. Many of these are subsets of our system, to justify the inclusion of additional complexity.

Similar Fact Count This baseline judges the truth of a fact by tuning a threshold on the total number of similar facts in the database. This baseline would perform well if our negative facts were noticeably disconnected from our database.

Argument Similarity A key discriminating feature may be the similarity between a_1 and a_2 in true versus false facts. This baseline thresholds the cosine similarity between arguments, tuned on the training data to maximize classification accuracy.

Cosine Similarity At its core, our model judges the truth of a fact based on its similarity to facts in the database; we create a baseline to capture this intuition. For every candidate fact (differing in either an argument or the relation), we compute the cosine similarity between the query and the candidate, evaluated on the differing terms. This baseline outputs the maximum similarity between a query and any candidate; a threshold on this similarity is tuned on the training data to maximize classification accuracy.

System	**ReVerb**		**ConceptNet**	
	Train	Test	Train	Test
random	50.0	50.0	50.0	50.0
count	51.9	52.3	51.0	51.6
argsim	52.0	52.6	62.1	60.0
cos	71.4	70.6	71.9	70.5
system	**74.3**	**74.2**	**76.5**	**74.3**

Table 4: Classification accuracy for ReVerb and ConceptNet data. The three baselines are described above the line as described in Section 5.1; random chance would get an accuracy of 50%.

5.2 Automatic Evaluation

A natural way to evaluate our system is to use the same regime as our training, evaluating on held out facts. For both domains we train on a balanced dataset of 20,000 training and 10,000 test examples. Performance is measured in terms of classification accuracy, with a random baseline of 50%.

Table 4 summarizes our results. The similar fact count baseline performs nearly at random chance, suggesting that our sampled negative facts cannot be predicted solely on the basis of connectedness with the rest of the database. Furthermore, we outperform the cosine baseline, supporting the intuition that aggregating similarity metrics is useful.

To evaluate the informativeness of the confidence our system produces, we can allow our system to abstain from unsure judgments. Recall refers to the percentage of facts the system chooses to make a guess on; precision is the percentage of those facts which are classified correctly. From this, we can create a precision/recall curve – presented in Figure 2 for ReVerb and Figure 3 for ConceptNet. Our system achieves an area under

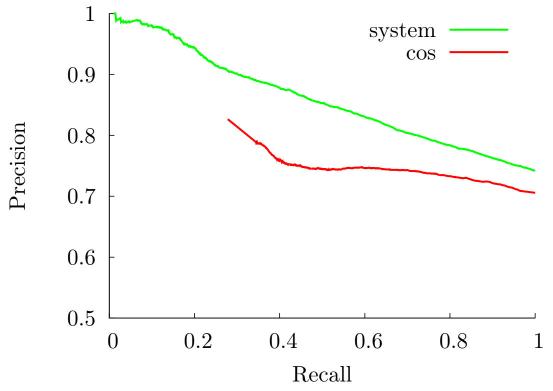

Figure 2: Accuracy of ReVerb classification, as a function of the percent of facts answered. The y axis begins at random chance (50%).

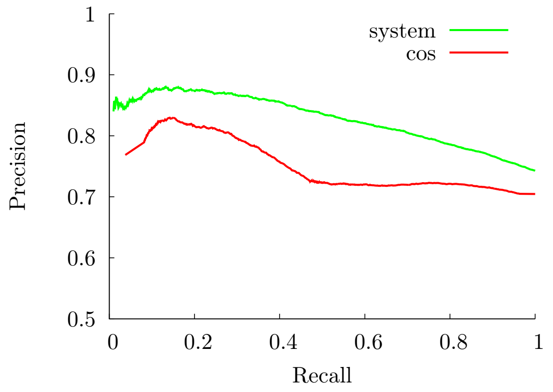

Figure 3: Accuracy of ConceptNet classification, as a function of the percent of facts answered. The y axis begins at random chance (50%).

the curve of 0.827 on ConceptNet (compared to the cosine baseline of 0.751). For ReVerb, we obtain an area of 0.860 (compared to 0.768 for the cosine baseline).[5]

5.3 ReVerb Filtering

In order to provide a grounded evaluation metric we evaluate our system as a confidence estimator for ReVerb extractions. Many ReVerb extractions are semantically implausible, or clash with common-sense intuition. We annotate a number of extractions on Mechanical Turk, and attempt to predict the extractions' feasibility.

This task is significantly more difficult than the intrinsic evaluations. Part of the difficulty stems

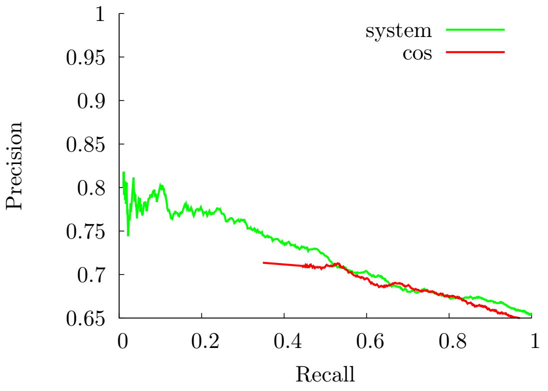

Figure 4: PR curve for ReVerb confidence estimation. The y axis of the graph is truncated at 65% – this corresponds to the majority class baseline.

from our database itself (and therefore our candidate similar facts) being unfiltered – our query facts empirically were and therefore in a sense *should* be in the database. Another part stems from these facts already having been filtered once by ReVerb's confidence estimator.

To collect training and test data, we asked workers on Amazon Mechanical Turk to rate facts as *correct*, *plausible*, or *implausible*. They were instructed that they need not research the facts, and that correct facts may be underspecified. Workers were given the following descriptions of the three possible responses:

- **Correct**: You would accept this fact if you read it in a reputable source (e.g., Wikipedia) in an appropriate context.
- **Plausible**: You would accept this fact if you read it in a storybook.
- **Implausible**: The fact is either dubious, or otherwise nonsense.

Below this, five examples were shown alongside one control (e.g., (*rock*, *float on*, *water*)). Workers who answered more than 20% of the controls incorrectly were discarded. In total, 9 workers and 117 of 1200 HITs were discarded.

Each example was shown to three separate workers; a final judgment was made by taking the majority vote between *correct* (corresponding to our notion of plausibility) and *implausible*, ignoring votes of *plausible*. In cases where all the votes were made for *plausible*, or there was a tie, the example was discarded.

The experiment was run twice on 2000 ReVerb extractions to collect training and test data. The

[5] Curves begin at the recall value given a system confidence of 1.0. For area under the curve calculations, this value is extended through to recall 0.

training corpus consists of 1256 positive and 540 negative examples (1796 total; 70% positive). The test corpus consists of 1286 positive and 689 negative examples (1975 total; 65% positive)

Our system was retrained with the human evaluated training data; to account for class bias, our system's classification threshold was then tuned on the training data, optimizing for area under the precision/recall curve. Figure 4 illustrates our results, bounded below by majority choice. Our system achieves an area under the curve of 0.721; the cosine baseline has an area of 0.696.

Our system offers a viable trade-off of recall in favor of precision. For example, keeping only a third of the data can reduce the error rate by 25% – this can be appealing for large corpora where filtering is frequent anyways.

5.4 Answer Validation Exercise

The Answer Validation Exercise, organized as a track at CLEF between 2006 and 2008, focuses on filtering candidate answers from question answering systems (Peñas et al., 2007; Peñas et al., 2008; Rodrigo et al., 2009). Systems are presented with a question, and a set of answers along with their justification. The answers are either validated, rejected, or given an annotation of unknown and ignored during scoring. Since the proportion of correct answers is small (around 10%), the evaluation measures precision and recall over true answers predicted by each system.

Many answers in the task are incorrect because they violate common-sense intuition – for instance, one answer to *What is leprosy?* was *Africa clinic*. While any such specific mistake is easy to fix, our approach can be a means of handling a wide range of such mistakes elegantly.

To adapt our system to the task, we first heuristically converted the question into a query fact using the subject and object Stanford Dependency labels (de Marneffe and Manning, 2008). If either the subject or object specifies a type (e.g., *Which **party** does Bill Clinton belong to?*), the score of the fact encoding this relationship (e.g., (*Democrat*, *be*, *party*)) is averaged with the main query. Next, answers with very little n-gram overlap between the justification and either the question or answer are filtered; this filters answers which may be correct, but were not properly justified. Lastly, our system trained on Turk data (see Section 5.3), predicts an answer to be correct if it scores above the 65th percentile of candidate response scores. Lastly, as our system has no principled way of handling numbers, any answer which is entirely numeric is considered invalid.

System	**2007**			**2008**		
	P	R	F_1	P	R	F_1
all validated	11	100	19	8	100	14
filter only	16	95	27	14	100	24
median	–	–	35	–	–	20
best	–	–	55	–	–	64
system	31	62	41	16	43	23

Table 5: Classification accuracy for the Answer Validation Exercise task. The baseline is accepting all answers as correct (*all validated*); a second baseline (*filter only*) incorporates only the n-gram overlap threshold. The median and top performing scores for both years are provided for comparison.

Results are shown in Table 5. We evaluate on the 2007 and 2008 datasets, outperforming the median score both years. Our system would place third out of the eight systems that competed in both the 2007 and 2008 tasks. As we are evaluating our system as a single component not trained on the task, we understandably fall well under the top performing systems; however, our performance is nonetheless an indication that the system provides a valuable signal for the task.

6 Conclusion

We have created a simple yet effective system to determine the plausibility of an arbitrary fact, both in terms of an intrinsic measure, and in downstream applications. Furthermore we have shown that the confidences returned by our system are informative, and that high-precision judgments can be obtained even at reasonable recall. We hope to devote future work to enriching the notion of fact similarity, and better handling the noise in the training data.

Acknowledgements We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) contract no. FA8750-13-2-0040. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of ACL*, Portland, OR.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. *Linguistic Data Consortium*.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, pages 13–47.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. The ClueWeb09 data set.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, pages 3–3.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545.

Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, pages 311–338.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL-HLT*, pages 541–550.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. *ACL*.

Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In *AAAI*, pages 1062–1068, Pittsburgh, PA.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430.

Hugo Liu and Push Singh. 2004. ConceptNet: a practical commonsense reasoning toolkit. *BT technology journal*, pages 211–226.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP*.

John McCarthy. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial intelligence*, pages 27–39.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, pages 39–41.

Judea Pearl. 1989. *Probabilistic semantics for non-monotonic reasoning: A survey*. Knowledge Representation and Reasoning.

Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. 2007. Overview of the answer validation exercise 2006. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, pages 257–264.

Anselmo Peñas, Álvaro Rodrigo, and Felisa Verdejo. 2008. Overview of the answer validation exercise 2007. In *Advances in Multilingual and Multimodal Information Retrieval*, pages 237–248.

Raymond Reiter. 1980. A logic for default reasoning. *Artificial intelligence*, pages 81–132.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL-HLT*, pages 74–84.

Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. 2009. Overview of the answer validation exercise 2008. In *Evaluating Systems for Multilingual and Multimodal Information Access*, pages 296–313.

Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *EMNLP*, pages 1088–1098.

Push Singh, Thomas Lin, Erik Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *ACL*, pages 801–808.

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Oren Etzioni, et al. 2010. Adapting open information extraction to domain-specific relations. *AI Magazine*, pages 93–102.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2011. Deriving a web-scale common sense fact database. In *AAAI*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*, pages 173–180.

Ellen M Voorhees. 2001. Question answering in TREC. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 535–537.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Probabilistic databases of universal schema. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 116–121.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *ACL-HLT*, pages 25–26.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, pages 658–666. AUAI Press.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687.