Deriving a Web-Scale Common Sense Fact Database

Niket Tandon and **Gerard de Melo** and **Gerhard Weikum**
Max Planck Institute for Informatics
Saarbrücken, Germany
{ntandon,demelo,weikum}@mpi-inf.mpg.de

Abstract

The fact that birds have feathers and ice is cold seems trivially true. Yet, most machine-readable sources of knowledge either lack such common sense facts entirely or have only limited coverage. Prior work on automated knowledge base construction has largely focused on relations between named entities and on taxonomic knowledge, while disregarding common sense properties. In this paper, we show how to gather large amounts of common sense facts from Web n-gram data, using seeds from the ConceptNet collection. Our novel contributions include scalable methods for tapping onto Web-scale data and a new scoring model to determine which patterns and facts are most reliable. The experimental results show that this approach extends ConceptNet by many orders of magnitude at comparable levels of precision.

Introduction

Motivation. Roses are red, violets are blue. Facts of this sort seem trivially true. Yet, knowledge that humans take for granted on a daily basis is not readily available in computational systems. For several decades, the *knowledge acquisition bottleneck* has been a major impediment to the development of intelligent systems. If such knowledge was more easily accessible, applications could behave more in line with users' expectations. For example, a mobile device could recommend nearby coffee shops rather than ice cream vendors when users desire warm beverages. A search engine would be able to suggest local supermarkets when a user wishes to buy soap. Further applications include query expansion (Hsu, Tsai, and Chen 2006), video annotation (Altadmri and Ahmed 2009), faceted search (Bast et al. 2007), and distance learning (Anacleto et al. 2006), among other things. Liu and Singh (2004) provide a survey of applications that have made use of explicit common sense facts.

Previous work to formalize our commonsense understanding of the world has largely been centered around

a) manual efforts, e.g. Cyc, SUMO, and WordNet, as well as resources like ConceptNet (Havasi, Speer, and Alonso 2007) that rely on crowd-sourcing,

b) minimally supervised information extraction from text (Etzioni et al. 2005; Suchanek, Sozio, and Weikum 2009; Carlson et al. 2010).

Both strategies are limited in scope or coverage. Human efforts are time-consuming and often fail to attract sufficient numbers of contributors. Information extraction methods have been successful for taxonomic knowledge (`IsA` and `InstanceOf` relationships among classes and between entities and classes), and for relations between named entities (e.g. birthplaces of people). Most extraction systems rely on pattern matching, e.g. a string like "… *cities such as Paris* … " matching the "*<X> such as <Y>*" pattern for the `IsA` relation leads to knowledge of the form `IsA(Paris,City)`. Previous work (Hearst 1992) has shown that textual patterns can be surprisingly reliable but are generally very rare. For instance, in a 20 million word New York Times article collection, Hearst found only 46 facts.

Contribution. This paper explores how large numbers of common sense properties like `CapableOf(dog,bark)`, `PartOf(room,house)` can be harvested automatically from the Web. A new strategy is proposed to overcome the robustness and scalability challenges of previous work.

- Rather than starting out with minimal numbers of seeds, we exploit information from an existing fact database, ConceptNet (Havasi, Speer, and Alonso 2007).
- Rather than using a text corpus, we rely on a Web-scale n-gram dataset, which gives us a synopsis of a significant fraction of *all* text found on the Web. While people rarely explicitly express the obvious, we believe that "*a word is characterized by the company it keeps*" (Firth 1957) and exploit the very large quantities of natural language text that are now available on the Web.
- Unlike standard bootstrapping approaches, we rely on novel scoring functions to very carefully determine which patterns are likely to lead to good extractions.
- Unlike previous unsupervised outputs, we rely on a semi-supervised approach for scoring the output facts. The model is obtained from the input data, without any need for additional manual labelling.

Related Work

Information Extraction. The idea of searching for occurrences of specific textual patterns in text to extract information has a long history. Patterns for finding `IsA` relationships were discussed theoretically (Lyons 1977; Cruse 1986) and later evaluated empirically (Hearst 1992). Since then, a large range of approaches have built upon these ideas, extending them to other relationships like `PartOf` (Girju, Badulescu, and Moldovan 2006) as well as factual knowledge like birth dates of people and capital cities (Cafarella et al. 2005).

To overcome the sparsity of pattern matches, iterative bootstrapping approaches attempt to re-use extraction results as seeds (Pantel and Pennacchiotti 2006). Unfortunately, the extraction quality often degrades very quickly after a few iterations. Our approach ensures that significant amounts of seeds and pattern matches are available in the first iteration, so additional rounds are not necessary.

Recent work (Suchanek, Sozio, and Weikum 2009) has used consistency constraints on fact hypotheses (e.g., among several birthplace candidates for a person, only one can be correct) to improve precision. However, these techniques are computationally much more expensive and it is an open issue if and how constraints could be formulated or learned for common sense properties. NELL (Carlson et al. 2010) relies on humans to filter the rules proposed by its rule learner.

Web-Scale Extraction. Most information extraction systems have to date only been evaluated on small corpora. Recent studies on scalable extraction (Pantel, Ravichandran, and Hovy 2004; Agichtein 2005) still relied on corpora that represent only very small fractions of the Web. Since Web-scale document collections are not easily obtainable, an alternative is to resort to using Web search engines (Etzioni et al. 2005; Schwartz and Gomez 2009). However, systems using search engines to discover new facts will generally only retrieve top-k results without being able to exploit the large amounts of facts in the long tail. For example, a query like "*such as*" cannot be used on its own to retrieve very large numbers of `IsA` facts. Approaches that use search engines to derive more information for specific output facts first need to obtain the set of candidates from some other source. Additionally, Cafarella et al. (2005) showed that using search engines is many orders of magnitude slower than relying on local document collections. Our approach avoids these problems by directly working with Web-scale n-gram statistics based on giga-scale numbers of documents.

Common Sense Knowledge Acquisition. Most previous work on common-sense knowledge acquisition has relied on human-supplied information (von Ahn, Kedia, and Blum 2006; Havasi, Speer, and Alonso 2007; Speer et al. 2009). Rather than explicitly soliciting contributions, we instead attempt to make use of the large amounts of information that humans have already implicitly revealed on the Web. Matuszek et al. (2005) used Web search engines to extend Cyc. There have been studies on applying hard-coded rules to parse trees of text (Schubert 2002; Clark and Harrison 2009), achieving a precision of around 50-70%.

Approach

N-Gram Synopses

One of the distinguishing features of our approach is our reliance on Web-scale n-gram datasets for knowledge acquisition, which can serve as a proxy for the entire World Wide Web, as opposed to using a much smaller text corpus. A word n-gram is a sequence of n consecutive word tokens in text. An n-gram dataset is a resource that, for a given n-gram $s = s_1 \cdots s_n$, provides the corresponding occurrence frequency $f(s)$ of that string in a large document collection. For example, $f($"*major cities like London*"$)$ would yield the number of times the string "*major cities like London*" occurs in the document collection. Additionally, we assume we can retrieve sets of matching n-grams $s \in f(q)$ for wildcard queries like $q =$ "*major cities like <X>*". Some of the available n-gram datasets (Wang et al. 2010) are computed from petabytes of text, yet such n-gram statistics have not been used in information extraction systems in previous work.

N-gram datasets have a couple of shortcomings with respect to conventional corpora. Due to the limited size of n-grams, one can generally only extract binary relationships between short of items of interest. These additionally need to be expressed using reasonably short patterns. Fortunately, most simple common sense facts fit this schema. We focus on binary predicates rather than axioms or know-how as captured in axiomatic knowledge bases like Cyc. Semantically, these are generalizations that would apply broadly but not necessarily universally, e.g. we may have both `HasProperty(apple,green)` and `HasProperty(apple,red)`.

The primary motivation for working with n-gram data is the sheer volume of information that they provide, which entails not only a greater coverage but can also provide additional evidence for increasing the precision. Due to the greater amount of redundancy, the system has more information to base its assessment on. Pantel, Ravichandran, and Hovy (2004) showed that scaling to larger text collections alone can allow a rather simple technique to outperform much more sophisticated algorithms.

Our approach will be to first gather a set of patterns that allow us to identify candidate facts. The n-gram frequency statistics for the candidate facts as occurring with specific patterns are then used to derive a vector representation for each candidate fact. Based on a training set of labelled facts, a learning algorithm finally determines which candidate facts should be accepted.

Candidate Pattern Induction

Our system begins with the pattern induction step, where it attempts to bootstrap the extraction starting out with just a set of correct seed instances of each relation under consideration. For instance, for the `PartOf` relation, it could use a list including `(finger,hand)`, `(leaves,trees)`, and `(windows,houses)`. For the `IsA` relation, seed patterns can include `(dogs,animals)` and `(gold,metal)`. The goal of this step is to obtain a list of simple textual patterns that can then be used to harvest further knowledge from the corpora.

We iterate over the n-gram dataset and look for n-grams that contain the two words that make up a seed. Given a match, we can obtain a pattern by replacing the seed words with wildcards and optionally pruning preceding and following words. For example, given a seed pair (dogs,animals) for the IsA relation, we may encounter an n-gram like "*with dogs and other animals*". This n-gram gives rise to two patterns: "*with <X>*$_{\text{NNS}}$ *and other <Y>*$_{\text{NNS}}$" and "*<X>*$_{\text{NNS}}$ *and other <Y>*$_{\text{NNS}}$", where NNS represents the part-of-speech tag of the words. Generally, we retain all words between the two seed words, as well as all combinations of the following:

- 0,...,n-2 preceding words before the first seed word
- 0,...,n-2 following words after the second seed word

If n-grams are restricted to a maximal length of n, there can hence be at most

$$\max_{x\in 1,\dots,n-1} x(n-x) = \max_{x\in\{\lfloor\frac{n}{2}\rfloor,\lceil\frac{n}{2}\rceil\}} x(n-x) = \left\lfloor \frac{1}{4}n^2 \right\rfloor$$

patterns per n-gram.

Judicious Pattern Assessment

In our approach, it is vital to score the patterns reliably. This has multiple reasons:

- Relying on a Web-scale n-gram dataset, we are faced with an enormous number of seed occurrences, and hence very large numbers of potential patterns.
- Unlike previous approaches that relied on around 5 seeds, we make use of the fact that significant efforts have already been made to collect instances of the most important common sense relations. We thus rely on a larger number of seeds taken from a knowledge base like ConceptNet. This is an advantage, but it also means that we find many more patterns than in previous studies.
- Applying *all* of these candidate patterns for extraction would lead to tremendous scalability and noise challenges. Many patterns just coincidentally match certain seeds. For instance, a pattern like "*<X> and <Y>*" is very likely to match a number of seeds in ConceptNet. However, it is obvious that it is not a very reliable pattern. In an n-gram dataset derived from petabytes of data, the pattern will match many millions of n-grams but only few of the matches will correspond to any particular semantic relationship.

The challenge is thus to make judicious choices and keep only promising patterns. We base our decision on two important observations:

i) First of all, given the comparably large number of seeds that are available in ConceptNet, any pattern that only matches a single seed in the entire n-gram dataset is likely to be an unreliable pattern that does not really correspond to the target relation. It is interesting to note that many information extraction systems have relied on Pointwise Mutual Information (PMI) scores to prune the pattern list (Pantel and Pennacchiotti 2006). Unfortunately, PMI considers raw frequencies only, without emphasizing the number of *distinct* seeds matched. In particular, it turns out that the PMI formula tends to give these rare patterns that only coincidentally match a single seed the highest scores. Hence keeping only the highest-ranked patterns in terms of PMI would leave us with only useless patterns.

ii) At the same time, there are patterns that match multiple seeds, but that simultaneously match seeds from other relations, distinct from the target relation. PMI does not explicitly consider whether a pattern also matches other relations than a given target relation. The more this occurs, the greater indication we have that the pattern is not a reliable indicator of a specific relation, but just a broad generic pattern that happens to match many different word combinations.

We devised a score that simultaneously addresses both of these issues. With respect to aspect i), we shall see later on in Figure 1 that the number of seeds $s(x)$ matched by patterns follows a power-law $s(x) \approx ax^k$, where the majority of the patterns are in the long tail. For different relations, the dominating patterns have different numbers of seeds, and we empirically found that it is not possible to choose a threshold that works well for all relations. Instead, we observe that the magnitude of the slope at a particular position x characterizes to what degree a pattern is in the long tail. We hence use least squares linear regression with respect to $\log s(x) = k \log x + \log a$ to estimate k and a from the data. For a given seed count $s(x)$, we have $x \approx \left(\frac{s(x)}{a}\right)^{\frac{1}{k}}$, and therefore

$$\frac{d}{dx}s(x) \approx akx^{k-1} = ak\left(\frac{s(x)}{a}\right)^{\frac{k-1}{k}} \qquad (1)$$

characterizes the slope at $s(x)$. The more negative this value is, the more dominating the pattern.

For aspect ii), we compute a score as follows

$$\phi(R_i, p) = \sum_{R_j, j\neq i} \frac{|s(R_i,p)|}{|s(R_i)|} - \frac{|s(R_j,p)|}{|s(R_j)|}. \qquad (2)$$

This score considers the number of seeds $s(R_j, p)$ that the pattern matches from relations R_j (other than R_i) in comparison to the fraction of seeds it matches for R_i.

There is often a trade-off between the two aspects, as a score that matches many seeds will also be more likely to falsely match other relations. Given $\frac{d}{dx}s(x)$ from Equation 1 for a relation R_i and a pattern p, as well as $\phi(R_i, p)$ from Equation 2, we combine both scores conjunctively:

$$\theta(R_i, p) = \frac{e^{\phi(R_i,p)}}{1+e^{\phi(R_i,p)}} \cdot \frac{|\frac{d}{dx}s(x)|}{1+|\frac{d}{dx}s(x)|} \qquad (3)$$

This corresponds to normalizing the two scores $\log|\frac{d}{dx}s(x)|$ and $\phi(R_i, p)$ to $[0, 1]$ using the logistic function and then multiplying them, which entails that only patterns with good characteristics with respect to both aspects will obtain high ranks. We can thus choose a set of top-ranked patterns that are sufficiently significant to match a sufficient number of seeds but at the same time do not overgenerate large numbers of irrelevant tuples.

Fact Extraction and Assessment

After the first step, we have a set P of patterns that characterize a given relation. A given pattern $p \in P$ can be instantiated for specific candidate facts (x, y) as $p(x, y)$ to yield an

Algorithm 1 Web-Scale Knowledge Acquisition

```
procedure HARVEST(n-gram dataset $f$, seeds $S_i$ for relations $R_1, \ldots, R_m$, optional negative seeds $S_i^-$)
    $P_1, \ldots, P_m \leftarrow$ INDUCE_PATTERNS($f, S_1, \ldots, S_m$)        ▷ collect patterns $P_i$ for each relation
    $K_i \leftarrow \emptyset \quad \forall i$        ▷ candidate facts
    for all $s \in f(*)$ do        ▷ for all n-grams
        for all $i$ in $1, \ldots, m$ and $p \in P_i$ do        ▷ for all patterns
            if $s \in f(p)$ then $K_i \leftarrow K_i \cup \{(\mathrm{ARGX}(p,s), \mathrm{ARGY}(p,s))\}$        ▷ if $s$ matches $p$, the arguments form new fact candidates
    create labeled training sets $T_i$ with labels $l_{R_i(x,y)} \in \{-1,+1\}$ for $(x,y) \in T_i$        ▷ using $S_i$ and optionally $S_i^-$
    for all $i \in 1, \ldots, m$ do
        for all $(x,y) \in T_i$ do        ▷ create training vectors
            create training vector $\mathbf{v}_{R_i(x,y)}$ using patterns in $P_i$ and n-gram dataset $f$
        learn model $M_i$ from $\{(\mathbf{v}_{R_i(x,y)}, l_{R_i(x,y)}) \mid (x,y) \in T_i\}$        ▷ use learning algorithm
    $K_i \leftarrow \{(x,y) \in K_i \mid M_i(\mathbf{v}_{R_i(x,y)}) > 0.5\} \quad \forall i$        ▷ vectors for candidates are created using $P_i$ and $f$ and assessed using $M_i$
    return accepted facts $K_1 \ldots, K_m$        ▷ accepted facts as final output
procedure INDUCE_PATTERNS(n-gram dataset $f$, seeds $S_1, \ldots, S_m$)
    $P_i \leftarrow \emptyset \quad \forall i$        ▷ sets of patterns
    for all $s \in f(*)$ do        ▷ for all n-grams
        for all $i$ in $1, \ldots, m$ and $(x,y) \in S_i$ do
            if $s$ contains $x$ and $y$ then $P_i \leftarrow P_i \cup$ CREATEPATTERNS($s, x, y$)        ▷ replace $x$ and $y$ in $s$ with wildcards, prune text
    $P_i \leftarrow \{p \in P_i \mid \theta(R_i, p) > \theta_{\min}\} \quad \forall i$        ▷ prune using Equation 3
    return $P_1, \ldots, P_m$
```

n-gram string. For instance, a pattern like "<X> *is located in* <Y>" can be instantiated with a fact (Paris,France) to yield an n-gram "*Paris is located in France*". For such n-grams, we can then consult an n-gram dataset f to obtain frequency information $f(p(x,y))$.

Certainly, one could use the union of all facts found as the final output. Fortunately, in a large corpus like the Web, a given fact will frequently occur with more than one pattern, and we can apply more sophisticated ranking measures to obtain cleaner results. We proceed as follows. Given the set of seeds S_i for relations $R_1, \ldots, R_m$, we compute an $m \times m$ square similarity matrix as $M_{i,j} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$. For each relation R_i, we then use S_i as a set of positive examples. If the database provides any negative seeds S_i^-, these are used as negative examples. If too few negative seeds are available, we rely on $\bigcup\limits_{j \neq i, M_{i,j} < 0.5\%} S_j$ as a pool of additional negative examples. We sample from this pool until we have an equal number of positive and negative examples, which can be used in conjunction with supervised learning algorithms.

For a set of l patterns $P = \{p_1, \ldots, p_l\}$ and a given pair of words (x,y) for some relation, we produce an $(l+1)$-dimensional vector $\mathbf{v}_{(x,y)}$ with

$$v_{(x,y),0} = |\{f\,(p_i(x,y)) > 0 \mid i = 1, \ldots, l\}|,$$

$$v_{(x,y),i} = \begin{cases} 1 & f\,(p_i(x,y)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

for $i > 0$. Algorithm 1 gives the overall procedure for extracting information. We induce patterns, prune the pattern list, and then iterate over the n-grams to find candidate facts K_i for each relation R_i. We then use a learning algorithm to derive prediction models M_i for each relation. The models provide values $M_i(\mathbf{v}_{R_i(x,y)}) \in [0,1]$, where values over 0.5 mean that the pair is accepted.

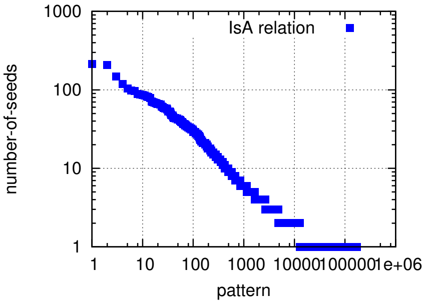

Figure 1: Number of seeds per pattern (log scale)

Table 1: Patterns for IsA (not showing POS tags), with <S>, </S> as sentence begin/end markers, respectively

Top-Ranked Patterns (PMI)	Top-Ranked Patterns (θ)
Varsity <Y> <X> Men	*<Y> / <X>*
<Y> MLB <X>	*<Y> : <X> </S>*
<Y> <X> Boys	*<Y> <X> </S>*
<Y> Posters <X> Basketball	*<Y> - <Y> </S>*
<Y> - <X> Basketball	*<Y> such as <X>*
<Y> MLB <X> NBA	*<S> <X> <Y>*
<Y> Badminton <X>	*<X> and other <Y>*

Results

Input Data

We use the following data sources:

- The Google Web 1T N-Gram Dataset Version 1 (Brants and Franz 2006): Google has published a dataset of raw frequencies for n-grams ($n = 1, \ldots, 5$) computed from over 1,024G word tokens of English text, taken from Google's Web page search index. In compressed form, the

Table 2: Overall Results

Relation	Prec˙	Recall	Final #Facts
`AtLocation`	57%	67%	13,273,408
`CapableOf`	77%	45%	907,173
`Causes`	88%	49%	3,218,388
`CausesDesire`	58%	61%	3,964,677
`ConceptuallyRelatedTo`	63%	60%	10,850,413
`CreatedBy`	44%	57%	274,422
`DefinedAs`	N/A	N/A	4,249,382
`Desires`	58%	65%	4,386,685
`HasA`	61%	64%	13,196,575
`HasFirstSubevent`	92%	86%	761,677
`HasLastSubevent`	96%	85%	797,245
`HasPainCharacter`	N/A	N/A	0
`HasPainIntensity`	N/A	N/A	0
`HasPrerequisite`	82%	55%	5,336,630
`HasProperty`	62%	48%	2,976,028
`HasSubevent`	54%	45%	2,720,891
`InheritsFrom`	N/A	N/A	106,647
`InstanceOf`	N/A	N/A	0
`IsA`	62%	27%	11,694,235
`LocatedNear`	71%	61%	13,930,656
`MadeOf`	52%	79%	13,412,950
`MotivatedByGoal`	53%	69%	76,212
`PartOf`	71%	58%	11,175,349
`ReceivesAction`	69%	70%	663,698
`SimilarSize`	74%	49%	8,640,737
`SymbolOf`	91%	64%	8,781,437
`UsedFor`	58%	49%	6,559,620

n-gram data amounts to 24GB. While the dataset does not include n-grams with a frequency of less than 40, the fact that it is distributed as a complete dataset means that additional post-processing and indexing can be applied to support a more sophisticated query syntax.

- ConceptNet 4.0 (2010-02 database): A database of facts generated from user contributions (Havasi, Speer, and Alonso 2007). Upon closer inspection, we discovered that ConceptNet is not as reliable as it could be. We found facts like `UsedFor(see, cut wood)` and `IsA(this, chair)`, resulting from misinterpretations of the user-provided sentences. Fortunately, our ranking scores make our approach robust with regard to inaccurate seeds.

Extraction

We used up to 200 seeds with a score of at least 3.0 in ConceptNet for each relation, with automatic addition of plural variants of words. The algorithm was implemented using Hadoop for distributed processing. Table 1 shows examples of top-ranked patterns for the `IsA` relation in terms of PMI and our θ (Eq. 3). Our analysis revealed several reasons why PMI is inadequate for Web-scale n-gram data:

- Influence of spam and boilerplate text: Large portions of the Web consist of automatically generated text, often replicated millions of times. PMI is misled by the high frequencies, whereas θ takes into account that such patterns only match few *distinct* seeds.

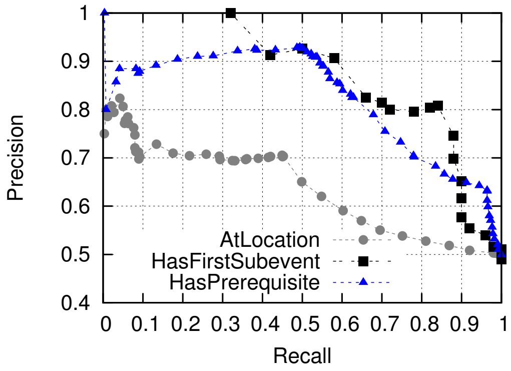

Figure 2: Precision-Recall Curve

- Less control over the selection of input documents: In n-gram datasets, there are large numbers of rare patterns that only coincidentally match some of the seeds.
- Less linguistically deep information, less context information: For instance, word boundaries and parts of speech may not be clear. Our own approach thus combines evidence from multiple patterns for scoring rather than trusting occurrences of any individual pattern.

Figure 1 shows the power law behaviour of the number of seeds per pattern, which leads to Equation 1. We empirically settled on a threshold of $\theta_{\min} = 0.6$, because lower thresholds were leading to overwhelming volumes of extraction data. This reduced the number of patterns to below 1000 per relation. In the fact extraction phase, we used the Stanford tagger to check part-of-speech tags when matching patterns.

Accuracy and Coverage

For the fact assessment, we relied on C4.5 Decision Trees with Adaptive Boosting (M1) to prune the output. Table 2 provides the final results. Precision and recall were computed using 10-fold leave-one-out cross-validation on the labelled sets T_i, which contained several thousand of human-supplied positive and negative examples from ConceptNet. For a small number of relations, there were insufficient seeds to find patterns or perform 10-fold cross-validation, but for most relations in ConceptNet, very encouraging results are obtained. Since the labelled sets are balanced, a random baseline would have only 50% precision. Additionally, we can opt to generate output of higher quality by trading off precision and recall and still obtain very large numbers of output facts. Figure 2 provides the precision-recall curve for three relations. We additionally verified the quality by manually assessing 100 random samples each of the accepted outputs for `CapableOf` (64% accuracy), `HasProperty` (78%), and IsA (67% accuracy).

The last column provides the final output results after classification retaining only those facts with decision tree leaf probabilities greater than 50%. We see that our resource is orders of magnitude larger than ConceptNet. Note that with a little additional supervision, the quality can be improved even further, e.g. in an active learning setting.

Pattern Clustering. We additionally experimented with Modularity Based Clustering (Clauset, Newman, and Moore 2004), using the bipartite graph between patterns and the extracted facts for the HasProperty relation. We obtained a modularity value of 0.61. Values of at least 0.3 indicate significant community structure. The resulting clusters can hence be useful for paraphrasing applications, e.g. "*the <X> is <Y>*" and "*the <X> was <Y>*" were clustered together.

Conclusion

We have introduced a framework for deriving a common sense fact database from ConceptNet in conjunction with Web-scale n-gram datasets based on tera-scale numbers of words. Although the overall recall is low relative to what is theoretically available on the Web, we are able to extend ConceptNet by many orders of magnitude.

In future work, we would also like to investigate coarse-grained word sense disambiguation approaches to distinguish senses of ambiguous words. Additionally, so far, we have only considered single tokens as names. We hope to extend our work to short multi-token units like "*mobile phone*" and "*meet people*". Finally, we would like to extend our setup to include additional relations. Incorporating WordNet (Fellbaum 1998), for instance, should straightforwardly be possible, while open information extraction, where the set of relations is unbounded, remains to be investigated. We hope that our database can pave the way for an entire ecosystem of novel intelligent applications.

References

Agichtein, E. 2005. Scaling information extraction to large document collections. *IEEE Data Eng. Bulletin* 28:3–10.

Altadmri, A. A., and Ahmed, A. A. 2009. VisualNet: Commonsense knowledgebase for video and image indexing and retrieval application. In *Proc. IEEE ICICS 2009, China*.

Anacleto, J. C.; de Carvalho, A. F. P.; de Almeida Néris, V. P.; de Souza Godoi, M.; Zem-Mascarenhas, S.; and Neto, A. T. 2006. How can common sense support instructors with distance education? In *Proc. SBIE 2006*.

Bast, H.; Chitea, A.; Suchanek, F.; and Weber, I. 2007. Ester: Efficient search in text, entities, and relations. In *Proc. SIGIR 2007*. Amsterdam, Netherlands: ACM.

Brants, T., and Franz, A. 2006. Web 1T 5-gram Version 1. *Linguistic Data Consortium, Philadelphia, PA, USA*.

Cafarella, M. J.; Downey, D.; Soderland, S.; and Etzioni, O. 2005. KnowItNow: Fast, scalable information extraction from the web. In *HLT/EMNLP*. ACL.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *Proc. AAAI 2010*.

Clark, P., and Harrison, P. 2009. Large-scale extraction and use of knowledge from text. In *Proc. K-CAP 2009*. ACM.

Clauset, A.; Newman, M.; and Moore, C. 2004. Finding community structure in very large networks. *Physical Review E* 70(6):66111.

Cruse, D. A. 1986. *Lexical Semantics*. Cambridge, UK: Cambridge University Press.

Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence* 165.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Firth, J. R. 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis* 1952-59:1–32.

Girju, R.; Badulescu, A.; and Moldovan, D. 2006. Automatic discovery of part-whole relations. *Computational Linguistics* 32(1):83–135.

Havasi, C.; Speer, R.; and Alonso, J. 2007. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Proc. RANLP 2007*.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. 14th COLING*, 539–545. Morristown, NJ, USA: ACL.

Hsu, M.-H.; Tsai, M.-F.; and Chen, H.-H. 2006. Query expansion with ConceptNet and WordNet: An intrinsic comparison. In *Information Retrieval Technology*.

Liu, H., and Singh, P. 2004. ConceptNet: a practical commonsense reasoning toolkit. *BT Technology Journal*.

Lyons, J. 1977. *Semantics, vol. 1*. Cambridge, UK: Cambridge University Press.

Matuszek, C.; Witbrock, M.; Kahlert, R.; Cabral, J.; Schneider, D.; Shah, P.; and Lenat, D. 2005. Searching for common sense: Populating Cyc from the Web. In *Proc. AAAI 1999*.

Pantel, P., and Pennacchiotti, M. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. ACL 2006*. ACL.

Pantel, P.; Ravichandran, D.; and Hovy, E. 2004. Towards terascale knowledge acquisition. In *Proc. COLING 2004*, 771. Morristown, NJ, USA: ACL.

Schubert, L. 2002. Can we derive general world knowledge from texts? In *Proc. HLT '02*.

Schwartz, H., and Gomez, F. 2009. Acquiring applicable common sense knowledge from the Web. In *Proc. NAACL HLT Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*. ACL.

Speer, R.; Krishnamurthy, J.; Havasi, C.; Smith, D.; Lieberman, H.; and Arnold, K. 2009. An interface for targeted collection of common sense knowledge using a mixture model. In *Proc. IUI 2009*, 137–146. ACM.

Suchanek, F. M.; Sozio, M.; and Weikum, G. 2009. SOFIE: a self-organizing framework for information extraction. In *Proc. WWW 2009*, 631–640. New York, NY, USA: ACM.

von Ahn, L.; Kedia, M.; and Blum, M. 2006. Verbosity: a game for collecting common-sense facts. In *Proc. CHI 2006*, 75–78. New York, NY, USA: ACM.

Wang, K.; Thrasher, C.; Viegas, E.; Li, X.; and Hsu, B. 2010. An overview of Microsoft Web N-gram corpus and applications. In *Proc. NAACL HLT 2010 Demonstration Session*. Los Angeles, CA, USA: ACL.