Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Acquiring Comparative Commonsense Knowledge from the Web

Niket Tandon
Max Planck Institute for Informatics
Saarbrücken, Germany
ntandon@mpi-inf.mpg.de

Gerard de Melo
IIIS,Tsinghua University
Beijing, China
demelo@tsinghua.edu.cn

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

Abstract

Applications are increasingly expected to make smart decisions based on what humans consider basic commonsense. An often overlooked but essential form of commonsense involves comparisons, e.g. the fact that bears are typically more dangerous than dogs, that tables are heavier than chairs, or that ice is colder than water. In this paper, we first rely on open information extraction methods to obtain large amounts of comparisons from the Web. We then develop a joint optimization model for cleaning and disambiguating this knowledge with respect to WordNet. This model relies on integer linear programming and semantic coherence scores. Experiments show that our model outperforms strong baselines and allows us to obtain a large knowledge base of disambiguated commonsense assertions.

Introduction

There is a growing conviction that the future of machine intelligence will crucially depend on our ability to exploit Big Data for more advanced human-like reasoning. Over the last decade, we have seen the rise of large knowledge collections driven by Big Data on the Web, most notably Wikipedia and online databases. Prominent examples include Freebase (Bollacker et al. 2008), YAGO (Hoffart et al. 2011), and DBpedia (Auer et al. 2007). These collections have enabled important advances in areas like Web Search and Question Answering, with Freebase forming the core of Google's Knowledge Graph and DBpedia and YAGO being used in IBM's Jeopardy!-winning Watson system. They also play an important role in intelligent dialog systems like Apple's Siri and Nuance's Nina, which are increasingly being adopted in mobile devices, television sets, automobiles, and automated customer service systems. Such systems however call for more than just typical factual knowledge like who starred in what movie or what the capital of Brazil is.

Increasingly, systems are expected to make smart commonsense choices. If the system suggests a pizza place and the user wants something "healthier", the computer should know that a seafood restaurant is probably a better suggestion than a burger joint. Rather than hard-coding various types of information into each relevant reasoning engine, genuine machine intelligence will require automatically drawing commonsense inferences from large amounts of data. Liu and Singh (2004) list a number of projects and applications that make use of common-sense knowledge bases.

In this paper, we focus on a novel aspect of this greater endeavor of making computer systems more intelligent. Our goal is to extract and infer large amounts of *comparative* knowledge about things in the world, e.g. that juice is sweeter than water, or that gold is more expensive than silver. This is an important part of human commonsense knowledge that has not been addressed by previous resources – neither by modern knowledge bases like Freebase nor by commonsense knowledge collections like Cyc (Matuszek et al. 2005), ConceptNet (Havasi, Speer, and Alonso 2007) or our own prior work WebChild (Tandon et al. 2014).

Our approach involves first using open information extraction (Open IE) techniques to harvest large amount of initial comparative data from the Web, e.g. the fact that seafood, on average, is perceived as healthier than a hamburger. Open IE leads to triples of surface phrases, e.g. (steel, sharper than, wood). Our method goes much further by computing triples of disambiguated word senses, e.g. (`steel-noun-2`, `sharper-than-adj-2`, `wood-noun-1`) or (`photo-noun-1`, `sharper-than-adj-1`, `sketch-noun-1`), where the numbers are the WordNet sense numbers for the ambiguous words.

Our method uses clustering and linear optimization methods to clean and consolidate this knowledge, while also inferring new information. In the end, we obtain sense-disambiguated knowledge that properly distinguishes, for example, the temperature sense of "*cool*" from the hipness sense of "*cool*".

Contributions. We make the following contributions.

1. We present the first open information extraction system for harvesting large amounts of *comparative knowledge* from Web contents.
2. We introduce a novel algorithm to organize such comparative knowledge with proper semantic rigor such that arguments are *sense-disambiguated*, by linking them to the lexical knowledge base WordNet (Fellbaum 1998)).
3. We publish a large, semantically refined knowledge base of comparative commonsense knowledge.

Overview

The goal of this paper is to establish a large machine-readable repository of comparative commonsense knowledge. In particular, we consider relationships that can be expressed using the comparative forms of adjectives, e.g. "is bigger than", "is more reliable than". As we are aiming at commonsense, most knowledge we would expect to capture will not hold as absolute universal truths, but rather be a reflection of overall tendencies. For example, although cheetahs are generally known to be faster than lions, an individual cheetah might be too young, unhealthy, etc. to be faster than a given lion.

Our input will be a large collection of text. Our output will be a set of annotated subject-predicate-object triples of the form (`X`, `ADJ`, `Y`). Here, `X` and `Y` are noun concepts, and `ADJ` is an adjective concept, interpreted as a comparative, e.g. (`car`, `faster`, `bike`), (`melon`, `bigger`, `apple`), (`lemon`, `more-sour`, `apple`)

The arguments we expect to obtain at the end are not ambiguous words but sense-specific identifiers for noun and adjective concepts. For this, we assume the existence of a repository of noun and adjective concept identifiers. Specifically, we rely on WordNet (Fellbaum 1998), a well-known lexical knowledge base that distinguishes the different senses of ambiguous words like "*bass*" (music instrument or fish) or "*green(er)*" (color or environmental friendliness), while also grouping together near-synonyms like "*fast(er)*" and "*quick(er)*".

In an effort to cover a wider range of commonsense phenomena, we do not limit ourselves to arguments `X` and `Y` that directly correspond to nominal concepts in WordNet. Additionally, we also aim at obtaining large amounts of information about more specific concepts as given by disambiguated adjective-noun or noun-noun combinations, e.g. young lions, cold water, vegetable dumpling, etc. We refer to these as *ad hoc concepts*.

Comparative Knowledge Base Construction

In order to arrive at such a knowledge base given just a large text collection, we (1) use information extraction to obtain observed textual facts, and then (2) develop a model to disambiguate and semantically organize this extracted data.

Open Information Extraction

In the extraction phase, we run through the input corpus and collect all triples matching the template *(noun phrase) + (comparative predicate) + (noun phrase)*.

As noun phrases, we consider nouns listed in WordNet ("*water*", "*dark chocolate*"), adjective-noun pairs ("*cold water*") and noun-noun expressions ("*football manager*") that are not in WordNet. These nouns phrases are stemmed after dropping any leading stop words ("*the*", "*a*", "*our*", etc.). We heuristically identify the head noun of a noun phrase containing an adjective or a noun word (ignoring proper nouns) as the right-most stemmed noun (water in cold water).

As comparative phrases, we allow inflections of the word "*to be*" followed by comparative forms of adjectives (e.g. "*bigger than*", "*more educated than*", etc.). We also allow them to contain modifying adverbs/negations, as e.g. in "*are typically bigger than*", "*are never bigger than*", or "*is not only faster than*". We manually encode a list of negation phrases like "*not*", "*never*" and some exceptions ("*not only*"). As a heuristic, we capture negations by assuming negations imply the opposite, in common-sense terms. Thus, "*bikes are not faster than cars*" is stored as a triple $(\mathrm{car}, \mathrm{faster}, \mathrm{bike})$. Comparative forms of adjectives are detected using WordNet. An exhaustive list of potential comparative forms of adjectives is generated by adding the suffix "er" and prefixes "more ", "less " to each WordNet adjective ("*colder*", "*more cold (than)*"). WordNet additionally provides a list of irregular forms that cannot be generated in this way (e.g. "*better*").

Using all of this information, we developed a fast pattern matcher to detect instances of this template. Our implementation is based on Hadoop MapReduce in order to quickly process large Web corpora in a distributed hardware cluster. The output of the extraction phase consists of i) left noun phrase (and its head noun), ii) relation (and its embedded adjective), iii) right noun phrase (and its head noun), iv) frequency, v) direction.

Disambiguation and Organization

The next step is to disambiguate and organize the knowledge. The original extractions are often ambiguous. For example, "*hotter than*" can refer to heat or to attractiveness, and "*richer than*" can refer to money or to calories. The left and right arguments are also often ambiguous. At the same time, our extractions do not group together equivalent forms. Given an original *observed triple* (n_1^*, a^*, n_2^*) from the information extraction step, our principal goal will be to choose relevant *grounded triples* (n_1, a, n_2), where n_1, a, and n_2 are no longer simple strings from the text, but disambiguated word sense IDs with respect to a lexical knowledge base like WordNet.

We first present a simple local baseline model, which assumes independence across the triples. Then we describe a more advanced model, which makes use of integer linear programming problems (ILPs) and does not assume independence across triples.

Local Model. Similar to state-of-the-art methods for word sense disambiguation on text (Navigli 2009), the local model assumes that the most likely disambiguation is the one that has the highest internal coherence, while simultaneously also preferring more frequent senses. A grounded triple exhibits high internal coherence when the word senses within it are similar to each other. Thus, for every grounding (n_1, a, n_2) of an observation (n_1^*, a^*, n_2^*), a score is computed as follows:

$$\begin{aligned}\mathrm{score}(n_1, a, n_2) \; = \; & \tau_{\mathrm{NN}}(n_1, n_2) \\ & + \tau_{\mathrm{NA}}(n_1, a) + \tau_{\mathrm{NA}}(n_2, a) \\ & + \phi(n_1^*, n_1) + \phi(n_2^*, n_2) \\ & + \phi(a^*, a) \end{aligned} \quad (1)$$

This score combines three different kinds of components:

- $\tau_{\mathrm{NN}}(n_1, n_2)$: A taxonomic relatedness score between two noun senses n_1 and n_2, computed using a WordNet path similarity measure (Pedersen, Patwardhan, and Michelizzi 2004). If one of the two arguments is an ad hoc concept like "*ripe fruit*", we have separate senses for the first word and for the second word, so we compute two scores and take the average. If both n_1 and n_2 are ad hoc concepts, we compute all four pairwise scores between involved senses for n_1 and senses for n_2, again taking the average. While doing this, any scores between two noun senses are computed as above using the WordNet path similarity measure, while any scores involving an adjective sense are computed as for $\tau_{\mathrm{NA}}(n, a)$ below.
- $\tau_{\mathrm{NA}}(n, a)$: A taxonomic relatedness score between a noun sense and an adjective sense, computed by determining the overlap between their extended WordNet glosses. The extended glosses are constructed by concatenating the original sense's gloss with the glosses of related senses in the taxonomic neighborhood. The taxonomic neighborhood of a sense includes its directly related senses (e.g. similar-to, antonym senses in WordNet). For nouns, the hypernyms and hyponyms of a given sense are also considered. We then create bag-of-words feature vectors and compute the cosine similarity.

  When n is an ad hoc concept, the relatedness is the average over the two scores between its respective component senses and the adjective a.
- $\phi(w, s)$: A prior for the sense s of a word w, computed as $\frac{1}{1+r}$, given the WordNet sense rank r. Thus, the first sense obtains $\frac{1}{2}$, the second sense $\frac{1}{3}$, and so on. For ad hoc concepts, the sense score is the average sense score of its components.

Joint Model. Although all of its components are well-motivated, the local model ultimately still only has a limited amount of information at its disposition. Two or more groundings can easily end up obtaining very similar scores, without a clear winner. In particular, the local model does not consider any form of dependency across grounded triples. In reality, however, the disambiguation of a triple like $(\mathrm{car}, \mathrm{faster}, \mathrm{bike})$ is highly correlated with the disambiguation of related triples, e.g. $(\mathrm{bicycle}, \mathrm{slower}, \mathrm{automobile})$. We thus designed a more sophisticated joint model based on the following desiderata.

a) Encourage high coherence within a triple and prefer frequent senses.

b) Encourage high coherence across chosen grounded triples.

c) Prefer same senses of a word across observations.

d) Properly handle ad hoc concepts.

We define our Joint Model using integer-linear programs (ILPs) to encode the intuition that similar grounded triples collectively aid in disambiguation. The desired properties are soft constraints and become part of the objective. We assume we are given a series of observed triples, denoted by index i. For each observed triple $(n_1^{i*}, a^{i*}, n_2^{i*})$, we have a number of candidate groundings, denoted by index j. We refer to such a grounded triple as $(n_1^{ij}, a^{ij}, n_2^{ij})$. The ILP requires precomputing the following coherence scores for such grounded triples.

- coh_{ij}: The coherence of an individual grounded triple, computed just like $\tau_{\mathrm{NN}}(n_1, n_2)+\tau_{\mathrm{NA}}(n_1, a)+\tau_{\mathrm{NA}}(n_2, a)$ in the local model.
- τ_{ij}: The average sense score of a grounded triple, computed as $\frac{1}{3}$ of $\phi(n_1^*, n_1) + \phi(n_2^*, n_2) + \phi(a^*, a)$ from the local model.
- $\mathrm{sim}_{ij,kl}$: The taxonomic relatedness between a grounded triple with index ij and another grounded triple with index kl. This is computed as

$$\sum_{i_1 \in \{1,2\}} \sum_{i_2 \in \{1,2\}} \tau_{\mathrm{NN}}(n_{i_1}^{ij}, n_{i_2}^{kl}) + \sum_{i_1 \in \{1,2\}} \tau_{\mathrm{NA}}(n_{i_1}^{ij}, a^{kl}) + \tau_{\mathrm{NA}}(n_{i_1}^{kl}, a^{ij}) + \tau_{\mathrm{AA}}(a^{ij}, a^{kl})$$

  where $\tau_{\mathrm{AA}}(a^{ij}, a^{kl})$ is a semantic relatedness score between two adjectives, computed as an extended gloss overlap just as for the τ_{NA} scores.
- $\mu_{ij,kl}$: Semantically equivalent triples are detected using synonymy and antonymy information, as explained later on in more detail. We set $\mu_{ij,kl} = 1$ if the two triples are semantically equivalent, and 0 otherwise.

Given these scores, our joint model relies on the objective and constraints provided in Table 1. In the objective function, the x_{ij} variables capture whether a given grounding is chosen and thus the first component encourages accepting groundings with high coherence and frequent senses, just like in the local model. The second component, in contrast, allows this model to go beyond the local model by encouraging that groundings are chosen that are similar to other chosen groundings. This is a major part of what allows our joint model to make joint decisions. We use $B_{ij,kl}$ variables to reflect whether two groundings were both simultaneously chosen. In practice, we prune the linear program significantly by only instantiating such variables when they are necessary. Finally, the third and fourth components encourage us to prefer fewer of the senses s of an adjective m or noun m, respectively, across the entire graph.

In order to ensure that all variables reflect their intended semantics, we need to enforce linear constraints. Constraint (1) specifies that a grounding can be either accepted or rejected. Constraint (2) ensures that at most one grounding of an observed triple is accepted. Note that the model does not require a grounding to be chosen for every observed triple. Constraints (3) to (5) ensure that the $B_{ij,kl}$ variables are 1 if and only if both x_{ij} and x_{kl} are 1.

Constraints (6) to (9) enforce that at least one word sense per word (adjective or noun, respectively) is accepted. Constraints (10) to (12) ensure that if a grounding is accepted, its word senses are marked as accepted.

Table 1: Joint Model Integer-Linear Program

maximize

$$\sum_i \sum_j (\text{coh}_{ij} + \tau_{ij}) x_{ij} \; + \; \sum_i \sum_j \sum_k \sum_l \text{sim}_{ij,kl} B_{ij,kl} \; - \sum_{m \in \text{adjectives}} \sum_s a_{ms} \; - \sum_{m \in \text{nouns}} \sum_s n_{ms}$$

subject to

x_{ij}	$\in$	$\{0,1\}$	$\forall i,j$	(1)
$\sum_j x_{ij}$	$\leq$	1	$\forall i,j$	(2)
$B_{ij,kl}$	$\in$	$\{0,1\}$	$\forall i,j,k,l$	(3)
$B_{ij,kl}$	$\leq$	x_{ij}	$\forall i,j,k,l$	(4)
$B_{ij,kl}$	$\leq$	x_{kl}	$\forall i,j,k,l$	(5)
a_{ms}	$\in$	$\{0,1\}$	$\forall m,s$	(6)
n_{ms}	$\in$	$\{0,1\}$	$\forall m,s$	(7)
$\sum_s a_{ms}$	$\geq$	1	$\forall$ adjectives m	(8)
$\sum_s n_{ms}$	$\geq$	1	$\forall$ nouns m	(9)
x_{ij}	$\leq$	a_{ms}	$\forall m,s$ of all adjective senses for i,j	(10)
x_{ij}	$\leq$	n_{ms}	$\forall m,s$ of all n_1 senses for i,j	(11)
x_{ij}	$\leq$	n_{ms}	$\forall m,s$ of all n_2 senses for i,j	(12)
x_{ij}	$=$	x_{kl}	$\forall i,j,k,l : \mu_{ij,kl} = 1$	(13)

Finally, constraint (13) ensures that semantically equivalent triples are tied together. Thus if one grounding is chosen, then all equivalents must be accepted as well. The model must either choose all or none of them. The details of how we determine $\mu_{ij,kl}$ are explained below in the next section.

Maximizing the objective subject to the constraints and taking those groundings for which $x_{ij} = 1$, we obtain a set of disambiguated triples that are not only highly ranked on their own but also coherent with the groundings chosen for related observations.

Triple Organization

In an effort to obtain a more well-defined and structured knowledge base, all semantically equivalent groundings are grouped together. For example, for appropriate chosen senses, the grounded triples (`car`, `faster`, `bike`), (`bicycle`, `slower`, `automobile`), (`car`, `speedier`, `cycle`) all express the fact that cars are generally faster than bicycles. We refer to a set of equivalent groundings as a *csynset*, similar to the notion of synsets (synonym sets) in WordNet. We use this notion of semantic equivalence for the $\mu_{ij,kl}$ scores in the ILP, to ensure consistency and joint assignments, as well as to provide the final output of our system in a more semantically organized form.

To determine equivalent triples, we make use of the following heuristics:

- Synonymy: Since groundings are disambiguated to WordNet synsets, groundings with synonymous word senses become identified, e.g. (`car`, `faster`, `bike`) and (`automobile`, `speedier`, `bicycle`).
- Antonymy: WordNet marks pairs of word senses like "*fast*" vs. "*slow*" as antonymous, i.e. as expressing semantically opposite meanings. If two adjective senses have opposite meanings, we can assume that their triples are equivalent if the arguments are in reverse order but otherwise equivalent. Thus (`car`, `faster`, `bike`) is equivalent to (`bike`, `slower`, `car`). Since WordNet's coverage of antonyms is limited, we also include indirect antonyms, considering antonymy for up to two levels of indirection (e.g. the synonym of an antonym of a synonym is also considered an antonym).
- Negation: While negation does not necessarily explicitly express the opposite, we have found that we obtain good results by treating negated adjectives just like antonyms. We use a small manually compiled list of negation markers for this.

Thus, our overall output is a large number of csynsets expressing comparative knowledge. Every csynset is itself a small set of equivalent grounded triples chosen by our joint model. To make our knowledge base more consistent, we finally check for any csynsets whose inverses are also present. In such cases, we sum up the frequencies of all observed triples belonging to the csynset, and keep only the direction with the higher frequency. This gives us our final output knowledge base, disambiguated and connected to WordNet.

Experiments

Corpora. We ran our extraction system on the following two very large Web corpora.

- **ClueWeb09:** The ClueWeb09 data set[1] is a large multilingual set of Web pages crawled from the Web in 2009. We used the 504 million Web pages in the English portion.
- **ClueWeb12:** The ClueWeb12 data set[2] consists of 27 TB of data from 733 million English Web pages crawled from the Web in 2012.

[1] http://lemurproject.org/clueweb09/
[2] http://lemurproject.org/clueweb12/

Table 2: Test Set Results (Precision)

Approach	WN	WN/ad hoc	ad hoc	all
MFS	0.42 ± 0.09	0.43 ± 0.09	0.46 ± 0.08	0.43 ± 0.05
Local Model	0.47 ± 0.09	0.49 ± 0.09	0.44 ± 0.08	0.47 ± 0.09
Joint Model	0.83 ± 0.06	0.85 ± 0.06	0.80 ± 0.06	0.83 ± 0.04

Table 3: Example Disambiguated Assertions

Type	Argument 1	Relation/Adjective	Argument 2
WN	snow-n-2	less dense-a-3	rain-n-2
	marijuana-n-2	more dangerous-a-1	alcohol-n-1
	diamond-n-1	sharper (sharp-a-3)	steel-n-2
WN/ad hoc	little child-n-1	happier (happy-a-1)	adult-n-1
	private_school-n-1	more expensive-a-1	public institute-n-1
	pot soil-n-3	heavier (heavy-a-1)	peat-n-1
ad hoc	peaceful resistance-n-1	more effective-a-1	violent resistance-n-1
	hot food-n-2	more delicious-a-2	cold dish-n-2
	wet wood-n-1	softer (soft-a-1)	dry wood-n-1

Evaluation dataset. Our extraction procedure compiled 488,055 triples from ClueWeb09 and 781,216 triples from ClueWeb12. To evaluate our system, we created three test sets sampling three different kinds of triples from this raw, ambiguous data:

i) **WN:** both left and right argument of the triple are surface forms that appear as words in WordNet, e.g. steel, wood, photo, sketch.

ii) **Ad-hoc:** both arguments are surface forms not contained in WordNet, e.g. math professor, novice student, digital image, brush sketch.

iii) **WN/ad-hoc:** one of the two arguments is in WordNet, the other is not.

Each of these three sample sets contained 100 randomly chosen observations. For each observation triple, human annotators were asked to choose the best possible word senses, not just surface forms. When an annotator found that none of the senses yields a true statement, i.e. the extracted triple is noise, then no senses were selected. In case of an ad hoc concept, the annotators annotated only the head word, e.g. (math professor (professor-noun-1), sharper than (sharp-adj-3), novice student (student-noun-1)). Our algorithm treats ad hoc concepts in the same way, disambiguating only the head-word.

Finally, we additionally relied on a separate set of around 40 annotated observations used for development and tuning, in order to avoid experimenting with different variants of our model on the test set.

Baselines. We consider the following two baselines.

1. **Most-frequent-sense heuristic (MFS):** The standard baseline for disambiguation tasks is MFS: use the Open IE surface-form triples and map them to the most frequent senses (using WordNet frequency information). In WordNet and many other lexical resources, sense entries are ranked such that the most frequent or important senses are listed first. For example, MFS disambiguates (car,fast,bike) as $(\texttt{car-n-1}, \texttt{fast-a-1}, \texttt{bike-n-1})$. In word sense disambiguation studies, the MFS heuristic has often been mentioned as hard to beat.
2. **Local model:** Our second baseline is the local model described earlier. For every observed triple, the top-ranked grounding with respect to the score from Eq. 1 is selected. This model thus not only uses the sense rankings but additionally incorporates the intra-grounding coherence. Unlike our joint model, however, this baseline disregards any coherence across triples.

Results. Having run our extraction code over the ClueWeb corpora, we obtained 488,055 extractions from ClueWeb09, and, 781,216 from ClueWeb12. Together, these amount to 1,064,066 distinct extractions. This is mainly due to the fact that the crawling strategies for the two ClueWeb corpora differed significantly. ClueWeb12 was created as a companion for ClueWeb09 with very different content (highly popular sites and Twitter links) and better spam detection. Thus there is little overlap between the two corpora.

In order to evaluate our joint model, we added additional related triples from the extractions to create a graph for every observed triple to be assessed. We greedily chose the most similar observed triples up to a maximal size of 10 observed triples, and then for every observed triple, possible candidate groundings were considered. We used these to instantiate the ILP, but smartly pruned out unnecessary variables (removing $B_{ij,kl}$ variables when $sim_{ij,kl}$ is zero or near-zero). For optimization, we use the Gurobi Optimizer package (Optimization 2014).

The evaluation is done separately for the three kinds of triples. Table 2 provides accuracy scores (95% Wilson confidence intervals) for the three different categories in the test

set, and for the overall test set aggregated over all the categories.

We see that the local model outperforms the most-frequent-sense baseline by a small margin. Although the local model makes use of valuable sense ranking and coherence information, it does not deliver satisfactory results. For example, the local model failed on (tiger, fast, auto) by incorrectly disambiguating it onto $(\texttt{tiger-n-1}(wrongsense : strongperson), \texttt{fast-a-1}, \texttt{auto-n-1})$.

Instead, our joint ILP model is the clear winner here, as it is able to take into account additional information about other related triples (e.g. car, slow, cheetah) when making decisions. As another example, given the observed triple $(\text{pork}, \text{more tender}, \text{beef})$, our model correctly infers that the highly ambiguous adjective "*tender*", with eight senses in Wordnet, is not used in its initial senses (sentiment-related) but in its fifth sense (easy to cut or chew). Our model simultaneously also correctly infers that "*pork*" is used in its first out of two listed senses, but that "*beef*" is not used in its first sense (cattle reared for their meat), but in its second out of three senses (meat from an adult domestic bovine).

Overall, our knowledge base provides around a million disambiguated comparative assertions. Table 3 lists some examples of the type of semantically refined knowledge one can find among these assertions.

Use Case. As an example use case, we consider computational advertisement, following Xiao and Blat (2013). Advertising frequently relies on metaphors to convey attributes of products and services. The salience of an attribute is typically manifested very well in comparative statements. For example, with smartness as the target attribute, we can query our knowledge base for triples with `smarter` as the relationship and obtain `dog-n-1`, `dolphin-n-1`, `pundit-n-1` as the most frequent left or right arguments. Similarly, with `heavier` as the relationship, the top three arguments are `iron-n-1`, `air-n-1`, `steel-n-1`.

Related Work

Knowledge Harvesting. Over the years, there have been numerous approaches to extract relationships between words from text, typically using string patterns or syntactic patterns. While some approaches aimed at lexical and commonsense relationships like hyponymy (Hearst 1992), part-whole relationships (Girju, Badulescu, and Moldovan 2006), and so on (Tandon, de Melo, and Weikum 2011), others focused more on encyclopedic facts like birth locations or company acquisitions (Agichtein and Gravano 2000). In particular, most of the recent large-scale knowledge bases like Freebase (Bollacker et al. 2008), DBpedia (Auer et al. 2007), and YAGO (Hoffart et al. 2011) typically only cover encyclopedic facts.

Additionally, all of the aforementioned approaches focus on a restricted manually pre-defined set of relationships. In our work, we aim to capture arbitrary comparative relationships expressed in text, in an open-ended manner. While several open information extraction have been presented in recent years (Etzioni et al. 2011; Carlson et al. 2010), existing systems simply deliver textual extractions rather than semantically disambiguated arguments. Thus, they might for example conflate the animal and car senses of "*Jaguar*". In order to move from the original text strings to more semantically coherent relationships, our approach relies on word sense disambiguation and classification techniques to consolidate equivalent extractions as well as disambiguate the arguments of each relation predicate.

Commonsense Knowledge. Among the efforts to build larger commonsense knowledge collections, one of the most well-known projects is the Cyc knowledge base (Matuszek et al. 2005), which was built by a large team of humans over the course of many years starting from 1984. However, Cyc focuses on general logical axioms rather than more specific commonsense facts about objects in the world. Another notable commonsense collection is ConceptNet (Havasi, Speer, and Alonso 2007), which consists mainly of crowdsourced information. However, ConceptNet's triples are not properly disambiguated and its limited set of relationship types (`causes`, `hasProperty`, `madeOf`, etc.) does not cover any comparative knowledge. WebChild (Tandon et al. 2014) is a recent large-scale commonsense knowledge base of sense-disambiguated object properties, but again lacks comparative knowledge. Thus our work fills an important gap.

There has been only little prior work on comparative knowledge mining and comparative commonsense knowledge mining in particular. Jain and Pantel (2011) used query logs to identify entities that are comparable in some respect, e.g. "Nikon D80 vs. Canon Rebel XT". Jang, Park, and Hwang (2012) proposed graph-based methods to predict new comparable entity pairs that have not been observed in the input data. Both of these approaches only produce sets of related entities and do not aim at gathering assertions about how they compare.

Jindal and Liu (2006) focused on the problem of identifying potential comparative sentences and their elements using sequence mining techniques. Such techniques could potentially be used to improve our extraction phase. In our setup, the extraction phase is not a major focus and thus we currently rely on simple heuristics that can easily be applied to terabyte-scale Web corpora. In the realm of commonsense knowledge, Cao et al. (2010) performed a small study on extracting commonsense comparative observations using manually defined patterns. The focus of our paper, in contrast, is to go beyond just a simple and small-scale extraction of natural language comparisons. Instead, we i) target large-scale ClueWeb data and ii) produce a large semantically disambiguated and consolidated knowledge base by recognizing semantically equivalent triples and using joint inference techniques.

Conclusion

We have presented the first approach for mining and consolidating large amounts of comparative commonsense knowledge from Big Data on the Web. Our algorithm successfully exploits dependencies between triples to connect and disambiguate the data, outperforming strong baselines by a large margin. The resulting knowledge base is freely available

from http://resources.mpi-inf.mpg.de/yago-naga/webchild/.

References

Agichtein, E., and Gravano, L. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, 85–94. New York, NY, USA: ACM.

Auer, S.; Bizer, C.; Lehmann, J.; Kobilarov, G.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A nucleus for a web of open data. In *Proc. ISWC/ASWC*, LNCS 4825. Springer.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, 1247–1250. New York, NY, USA: ACM.

Cao, Y.; Cao, C.; Zang, L.; Wang, S.; and Wang, D. 2010. Extracting comparative commonsense from the web. 340:154–162.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. H.; and Mitchell, T. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 1306–1313. AAAI Press.

Etzioni, O.; Fader, A.; Christensen, J.; Soderland, S.; and Mausam. 2011. Open information extraction: The second generation. In Walsh, T., ed., *IJCAI*, 3–10. IJCAI/AAAI.

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

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.

Hoffart, J.; Suchanek, F.; Berberich, K.; Lewis-Kelham, E.; de Melo, G.; and Weikum, G. 2011. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. In Srinivasan, S.; Ramamritham, K.; Kumar, A.; Ravindra, M. P.; Bertino, E.; and Kumar, R., eds., *Proceedings of the 20th International World Wide Web Conference (WWW2011) - Companion Volume*. New York, NY, USA: ACM.

Jain, A., and Pantel, P. 2011. How do they compare? automatic identification of comparable entities on the Web. *2011 IEEE International Conference on Information Reuse & Integration* 228–233.

Jang, M.; Park, J.-w.; and Hwang, S.-w. 2012. Predictive mining of comparable entities from the web.

Jindal, N., and Liu, B. 2006. Mining comparative sentences and relations. 1331–1336.

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

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*.

Navigli, R. 2009. Word Sense Disambiguation: a survey. *ACM Computing Surveys* 41(2):1–69.

Optimization, G. 2014. Inc. gurobi optimizer reference manual, version 5.6.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, 38–41. Association for Computational Linguistics.

Tandon, N.; de Melo, G.; Suchanek, F.; and Weikum, G. 2014. Webchild: Harvesting and organizing commonsense knowledge from the web. In *Proceedings of ACM WSDM 2014*.

Tandon, N.; de Melo, G.; and Weikum, G. 2011. Deriving a web-scale commonsense fact database. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*. Palo Alto, CA, USA: AAAI Press.

Xiao, P., and Blat, J. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity*.