Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

ConceptNet 5.5: An Open Multilingual Graph of General Knowledge

Robyn Speer
Luminoso Technologies, Inc.
675 Massachusetts Avenue
Cambridge, MA 02139

Joshua Chin
Union College
807 Union St.
Schenectady, NY 12308

Catherine Havasi
Luminoso Technologies, Inc.
675 Massachusetts Avenue
Cambridge, MA 02139

Abstract

Machine learning about language can be improved by supplying it with specific knowledge and sources of external information. We present here a new version of the linked open data resource ConceptNet that is particularly well suited to be used with modern NLP techniques such as word embeddings.

ConceptNet is a knowledge graph that connects words and phrases of natural language with labeled edges. Its knowledge is collected from many sources that include expert-created resources, crowd-sourcing, and games with a purpose. It is designed to represent the general knowledge involved in understanding language, improving natural language applications by allowing the application to better understand the meanings behind the words people use.

When ConceptNet is combined with word embeddings acquired from distributional semantics (such as word2vec), it provides applications with understanding that they would not acquire from distributional semantics alone, nor from narrower resources such as WordNet or DBPedia. We demonstrate this with state-of-the-art results on intrinsic evaluations of word relatedness that translate into improvements on applications of word vectors, including solving SAT-style analogies.

Introduction

ConceptNet is a knowledge graph that connects words and phrases of natural language (*terms*) with labeled, weighted edges (*assertions*). The original release of ConceptNet (Liu and Singh 2004) was intended as a parsed representation of Open Mind Common Sense (Singh 2002), a crowd-sourced knowledge project. This paper describes the release of ConceptNet 5.5, which has expanded to include lexical and world knowledge from many different sources in many languages.

ConceptNet represents relations between words such as:

- A *net* is used for *catching fish*.
- "*Leaves*" is a form of the word "*leaf*".
- The word *cold* in English is *studený* in Czech.
- O *alimento* é usado para *comer* [Food is used for eating].

In this paper, we will concisely represent assertions such as the above as triples of their start node, relation label, and end node: the assertion that "a dog has a tail" can be represented as (*dog*, *HasA*, *tail*).

ConceptNet also represents links between knowledge resources. In addition to its own knowledge about the English term *astronomy*, for example, ConceptNet contains links to URLs that define *astronomy* in WordNet, Wiktionary, OpenCyc, and DBPedia.

The graph-structured knowledge in ConceptNet can be particularly useful to NLP learning algorithms, particularly those based on word embeddings, such as (Mikolov et al. 2013). We can use ConceptNet to build semantic spaces that are more effective than distributional semantics alone.

The most effective semantic space is one that learns from both distributional semantics and ConceptNet, using a generalization of the "retrofitting" method (Faruqui et al. 2015). We call this hybrid semantic space "ConceptNet Numberbatch", to clarify that it is a separate artifact from ConceptNet itself.

ConceptNet Numberbatch performs significantly better than other systems across many evaluations of word relatedness, and this increase in performance translates to improvements on downstream tasks such as analogies. On a corpus of SAT-style analogy questions (Turney 2006), its accuracy of 56.1% outperforms other systems based on word embeddings and ties the previous best overall system, Turney's LRA. This level of accuracy is only slightly lower than the performance of the average human test-taker.

Building word embeddings is not the only application of ConceptNet, but it is a way to apply ConceptNet that achieves clear benefits and is compatible with ongoing research in distributional semantics.

After introducing related work, we will begin by describing ConceptNet 5.5 and its features, show how to use ConceptNet alone as a semantic space and a measure of word relatedness, and then proceed to describe and evaluate the hybrid system ConceptNet Numberbatch on these various semantic tasks.

Related Work

ConceptNet is the knowledge graph version of the Open Mind Common Sense project (Singh 2002), a common sense

knowledge base of the most basic things a person knows. It was last published as version 5.2 (Speer and Havasi 2013).

Many projects strive to create lexical resources of general knowledge. Cyc (Lenat and Guha 1989) has built an ontology of common-sense knowledge in predicate logic form over the decades. DBPedia (Auer et al. 2007) extracts knowledge from Wikipedia infoboxes, providing a large number of facts, largely focused on named entities that have Wikipedia articles. The Google Knowledge Graph (Singhal 2012) is perhaps the largest and most general knowledge graph, though its content is not freely available. It focuses largely on named entities that can be disambiguated, with a motto of "things, not strings".

ConceptNet's role compared to these other resources is to provide a sufficiently large, free knowledge graph that focuses on the common-sense meanings of words (not named entities) as they are used in natural language. This focus on words makes it particularly compatible with the idea of representing word meanings as vectors.

Word embeddings represent words as dense unit vectors of real numbers, where vectors that are close together are semantically related. This representation is appealing because it represents meaning as a continuous space, where similarity and relatedness can be treated as a metric. Word embeddings are often produced as a side-effect of a machine learning task, such as predicting a word in a sentence from its neighbors. This approach to machine learning about semantics is sometimes referred to as *distributional semantics* or *distributed word representations*, and it contrasts with the knowledge-driven approach of semantic networks or knowledge graphs.

Two prominent matrices of embeddings are the word2vec embeddings trained on 100 billion words of Google News using skip-grams with negative sampling (Mikolov et al. 2013), and the GloVe 1.2 embeddings trained on 840 billion words of the Common Crawl (Pennington, Socher, and Manning 2014). These matrices are downloadable, and we will be using them both as a point of comparison and as inputs to an ensemble. Levy, Goldberg, and Dagan (2015) evaluated multiple embedding techniques and the effects of various explicit and implicit hyperparameters, produced their own performant word embeddings using a truncated SVD of words and their contexts, and provided recommendations for the engineering of word embeddings.

Holographic embeddings (Nickel, Rosasco, and Poggio 2016) are embeddings learned from a labeled knowledge graph, under the constraint that a circular correlation of these embeddings gives a vector representing a relation. This representation seems extremely relevant to ConceptNet. In our attempt to implement it on ConceptNet so far, it has converged too slowly to experiment with, but this could be overcome eventually with some optimization and additional computing power.

Structure of ConceptNet

Knowledge Sources

ConceptNet 5.5 is built from the following sources:

Figure 1: ConceptNet's browsable interface (conceptnet.io) shows facts about the English word "bicycle".

- Facts acquired from Open Mind Common Sense (OMCS) (Singh 2002) and sister projects in other languages (Anacleto et al. 2006)
- Information extracted from parsing Wiktionary, in multiple languages, with a custom parser ("Wikiparsec")
- "Games with a purpose" designed to collect common knowledge (von Ahn, Kedia, and Blum 2006) (Nakahara and Yamada 2011) (Kuo et al. 2009)
- Open Multilingual WordNet (Bond and Foster 2013), a linked-data representation of WordNet (Miller et al. 1998) and its parallel projects in multiple languages
- JMDict (Breen 2004), a Japanese-multilingual dictionary
- OpenCyc, a hierarchy of hypernyms provided by Cyc (Lenat and Guha 1989), a system that represents common sense knowledge in predicate logic
- A subset of DBPedia (Auer et al. 2007), a network of facts extracted from Wikipedia infoboxes

With the combination of these sources, ConceptNet contains over 21 million edges and over 8 million nodes. Its English vocabulary contains approximately 1,500,000 nodes, and there are 83 languages in which it contains at least 10,000 nodes.

The largest source of input for ConceptNet is Wiktionary, which provides 18.1 million edges and is mostly responsible for its large multilingual vocabulary. However, much of the character of ConceptNet comes from OMCS and the various games with a purpose, which express many different kinds of relations between terms, such as *PartOf* ("a wheel is part of a car") and *UsedFor* ("a car is used for driving").

Relations

ConceptNet uses a closed class of selected relations such as *IsA*, *UsedFor*, and *CapableOf*, intended to represent a relationship independently of the language or the source of the terms it connects.

ConceptNet 5.5 aims to align its knowledge resources on its core set of 36 relations. These generalized relations are similar in purpose to WordNet's relations such as *hyponym* and *meronym*, as well as to the qualia of the Generative Lexicon theory (Pustejovsky 1991). ConceptNet's edges are directed, but as a new feature in ConceptNet 5.5, some relations are designated as being symmetric, such as *SimilarTo*. The directionality of these edges is unimportant.

The core relations are:

- **Symmetric relations**: *Antonym*, *DistinctFrom*, *EtymologicallyRelatedTo*, *LocatedNear*, *RelatedTo*, *SimilarTo*, and *Synonym*
- **Asymmetric relations**: *AtLocation*, *CapableOf*, *Causes*, *CausesDesire*, *CreatedBy*, *DefinedAs*, *DerivedFrom*, *Desires*, *Entails*, *ExternalURL*, *FormOf*, *HasA*, *HasContext*, *HasFirstSubevent*, *HasLastSubevent*, *HasPrerequisite*, *HasProperty*, *InstanceOf*, *IsA*, *MadeOf*, *MannerOf*, *MotivatedByGoal*, *ObstructedBy*, *PartOf*, *ReceivesAction*, *SenseOf*, *SymbolOf*, and *UsedFor*

Definitions and examples of these relations appear in a page of the ConceptNet 5.5 documentation[1].

Relations with specific semantics, such as *UsedFor* and *HasPrerequisite*, tend to connect common words and phrases, while rarer words are connected by more general relations such as *Synonym* and *RelatedTo*.

An example of edges in ConceptNet, in a browsable interface that groups them by their relation expressed in natural English, appears in Figure 1.

Term Representation

ConceptNet represents terms in a standardized form. The text is Unicode-normalized in NFKC form[2] using Python's `unicodedata` implementation, lowercased, and split into non-punctuation tokens using the tokenizer in the Python package `wordfreq` (Speer et al. 2016), which builds on the standard Unicode word segmentation algorithm. The tokens are joined with underscores, and this text is prepended with the URI `/c/lang`, where *lang* is the BCP 47 language code[3] for the language the term is in. As an example, the English term "United States" becomes `/c/en/united_states`.

Relations have a separate namespace of URIs prefixed with `/r`, such as `/r/PartOf`. These relations are given artificial names in English, but apply to all languages. The statement that was obtained in Portuguese as "O *alimento* é usado para *comer*" is still represented with the relation `/r/UsedFor`.

The most significant change from ConceptNet 5.4 and earlier is in the representation of terms. ConceptNet 5.4 required terms in English to be in lemmatized form, so that, for example, "United States" had to be represented as `/c/en/unite_state`. In this representation, "drive" and "driving" were the same term, allowing the assertions (*car*, *UsedFor*, *driving*) and (*drive*, *HasPrerequisite*, *have license*) to be connected. ConceptNet 5.5 removes the lemmatizer, and instead relates inflections of words using the *FormOf* relation. The two assertions above are now linked by the third assertion (*driving*, *FormOf*, *drive*), and both "driving" and "drive" can be looked up in ConceptNet.

[1]https://github.com/commonsense/conceptnet5/wiki/Relations

[2]http://unicode.org/reports/tr15/

[3]https://tools.ietf.org/html/bcp47

Vocabulary

When building a knowledge graph, the decision of what a node should represent has significant effects on how the graph is used. It also has implications that can make linking and importing other resources non-trivial, because different resources make different decisions about their representation.

In ConceptNet, a node is a word or phrase of a natural language, often a common word in its undisambiguated form. The word "lead" in English is a term in ConceptNet, represented by the URI `/c/en/lead`, even though it has multiple meanings. The advantage of ambiguous terms is that they can be extracted easily from natural language, which is also ambiguous. This ambiguous representation is equivalent to that used by systems that learn distributional semantics from text.

ConceptNet's representation allows for more specific, disambiguated versions of a term. The URI `/c/en/lead/n` refers to noun senses of the word "lead", and is effectively included within `/c/en/lead` when searching or traversing ConceptNet, and linked to it with the implicit relation *SenseOf*. Many data sources provide information about parts of speech, allowing us to use this as a common representation that provides a small amount of disambiguation. Further disambiguation is allowed by the URI structure, but not currently used.

Linked Data

ConceptNet imports knowledge from some other systems, such as WordNet, into its own representation. These other systems have their own target vocabularies that need to be aligned with ConceptNet, which is usually an underspecified, many-to-many alignment.

A term that is imported from another knowledge graph will be connected to ConceptNet nodes via the relation *ExternalURL*, pointing to an absolute URL that represents that term in that external resource. This newly-introduced relation preserves the provenance of the data and enables looking up what the untransformed data was. ConceptNet terms can also be represented as absolute URLs, so this allows ConceptNet to connect bidirectionally to the broader ecosystem of Linked Open Data.

Applying ConceptNet to Word Embeddings

Computing ConceptNet Embeddings Using PPMI

We can represent the ConceptNet graph as a sparse, symmetric term-term matrix. Each cell contains the sum of the weights of all edges that connect the two corresponding terms. For performance reasons, when building this matrix, we prune the ConceptNet graph by discarding terms connected to fewer than three edges.

We consider this matrix to represent terms and their contexts. In a corpus of text, the context of a term would be the terms that appear nearby in the text; here, the context is the other nodes it is connected to in ConceptNet. We can calculate word embeddings directly from this sparse matrix by following the practical recommendations of Levy, Goldberg, and Dagan (2015).

As in Levy et al., we determine the pointwise mutual information of the matrix entries with context distributional smoothing, clip the negative values to yield positive pointwise mutual information (PPMI), reduce the dimensionality of the result to 300 dimensions with truncated SVD, and combine the terms and contexts symmetrically into a single matrix of word embeddings.

This gives a matrix of word embeddings we call ConceptNet-PPMI. These embeddings implicitly represent the overall graph structure of ConceptNet, and allow us to compute the approximate connectedness of any pair of nodes.

We can expand ConceptNet-PPMI to restore the nodes that we pruned away, assigning them vectors that are the average of their neighboring nodes.

Combining ConceptNet with Distributional Word Embeddings

Having created embeddings from ConceptNet alone, we would now like to create a more robust set of embeddings that represents both ConceptNet and distributional word embeddings learned from text.

Retrofitting (Faruqui et al. 2015) is a process that adjusts an existing matrix of word embeddings using a knowledge graph. Retrofitting infers new vectors q_i with the objective of being close to their original values, $\hat{q}_i$, and also close to their neighbors in the graph with edges E, by minimizing this objective function:

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

Faruqui et al. give a simple iterative process to minimize this function over the vocabulary of the original embeddings.

The process of "expanded retrofitting" (Speer and Chin 2016) can optimize this objective over a larger vocabulary, including terms from the knowledge graph that do not appear in the vocabulary of the word embeddings. This effectively sets $\alpha_i = 0$ for terms whose original values are undefined. We set β_{ij} according to the weights of the edges in ConceptNet.

The particular benefit of expanded retrofitting to ConceptNet is that it can benefit from the multilingual connections in ConceptNet. It learns more about English words via their translations in other languages, and also gives these foreign-language terms useful embeddings in the same space as the English terms. The effect is similar to the work of Xiao and Guo (2014), who also propagate multilingual embeddings using crowd-sourced Wiktionary entries.

We add one more step to retrofitting, which is to subtract the mean of the vectors that result from retrofitting, then renormalize them to unit vectors. Retrofitting has a tendency to move all vectors closer to the vectors for highly-connected terms such as "person". Subtracting the mean helps to ensure that terms remain distinguishable from each other.

Combining Multiple Sources of Embeddings

Retrofitting can be applied to any existing matrix of word embeddings, without needing access to the data that was used to train them. This is particularly useful because it allows building on publicly-released matrices of embeddings whose input data is unavailable or difficult to acquire.

As described in the "Related Work" section, word2vec and GloVe both provide recommended pre-trained matrices. These matrices represent somewhat different domains of text and have complementary strengths, and the way that we can benefit from them the most is by taking both of them as input.

To do this, we apply retrofitting to both matrices, then find a globally linear projection that aligns the results on their common vocabulary. This process was inspired by Zhao, Hassan, and Auli (2015). We find the projection by concatenating the columns of the matrices and reducing them to 300 dimensions using truncated SVD. We then use this alignment to infer compatible embeddings for terms that are missing from one of the vocabularies.

In ongoing work, we are experimenting with additionally including distributional word embeddings from corpora of non-English text in this merger. Preliminary results show that this improves the multilingual performance of the embeddings.

After retrofitting and merging, we have a labeled matrix of word embeddings whose vocabulary is derived from word2vec, GloVe, and the pruned ConceptNet graph. As in ConceptNet-PPMI, we re-introduce all the nodes from ConceptNet by looking up and averaging their neighboring nodes.

Evaluation

To compare the performance of fully-built systems of word embeddings, we will first compare their results on intrinsic evaluations of word relatedness, then apply the word embeddings to the downstream tasks of solving proportional analogies and choosing the sensible ending to a story, to evaluate whether better embeddings translate to better performance on semantic tasks.

The hybrid system described above is the system we name ConceptNet Numberbatch, with the version number 16.09 indicating that it was built in September 2016. We now compare results from ConceptNet Numberbatch 16.09 to other systems that make their word embeddings available, both those that were used in building ConceptNet Numberbatch and a recently-released system, LexVec, that was not. The systems we evaluate are:

- word2vec SGNS (Mikolov et al. 2013), trained on Google News text
- GloVe 1.2 (Pennington, Socher, and Manning 2014), trained on the Common Crawl
- LexVec (Salle, Idiart, and Villavicencio 2016), trained on the English Wikipedia and NewsCrawl 2014
- ConceptNet-PPMI, described here and trained on ConceptNet 5.5 alone
- ConceptNet Numberbatch 16.09, the hybrid of ConceptNet 5.5, word2vec, and GloVe described here

Evaluations of Word Relatedness

One way to evaluate the intrinsic performance of a semantic space is to ask it to rank the relatedness of pairs of words, and compare its judgments to human judgments.[4] If one word in a pair is out-of-vocabulary, the pair is assumed to have a relatedness of 0. A good semantic space will provide a ranking of relatedness that is highly correlated with the human gold-standard ranking, as measured by its Spearman correlation (ρ).

Many gold standards of word relatedness are in common use. Here, we focus on MEN-3000 (Bruni, Tran, and Baroni 2014), a large crowd-sourced ranking of common words; RW (Luong, Socher, and Manning 2013), a ranking of rare words; WordSim-353 (Finkelstein et al. 2001), a smaller evaluation that has been used as a benchmark for many methods; and MTurk-771 (Halawi et al. 2012), another crowd-sourced evaluation of a variety of words.

To avoid manually overfitting by designing our semantic space around a particular evaluation, we experimented using smaller development sets, holding out some test data until it was time to include results in this paper:

- MEN-3000 is already divided into a 2000-item development set and a 1000-item test set. We use the results from the test set as the final results.
- RW has no standard dev/test breakdown. We sampled 2/3 of its items as a development set and held out the other 1/3 (every third line of the file, starting with the third).
- We used all of WordSim-353 in development. We examine its results both in English and in its Spanish translation (Hassan and Mihalcea 2009).
- We did not use MTurk-771 in development, holding out the entire set as a final test, showing that ConceptNet Numberbatch performs well on a previously-unseen evaluation.

We use the Spanish WordSim-353 as an example of a prominent non-English evaluation, indicating that expanded retrofitting is sufficient to learn vectors for non-English languages, even when all the distributional semantics takes place in English. However, a thorough multilingual evaluation is beyond the scope of this paper; the systems we compare to have only made English vectors available, and it would add considerable complexity to the evaluation to reproduce other systems of multilingual embeddings, accounting for their various ways of handling morphology and OOV words.

Solving SAT-style Analogies

Proportional analogies are statements of the form "a_1 is to b_1 as a_2 is to b_2". The task of filling in missing values of a proportional analogy was common until recently on standardized tests such as the SAT. Now, it is popular as a way to show that a semantic space can approximate relationships between words, even without taking explicit relationships into account.

Much of the groundwork for evaluating systems' ability to solve proportional analogies was laid by Peter Turney, including his method of Latent Relational Analysis (Turney 2006), which was quite effective at solving proportional analogies by repeatedly searching the Web for the words involved in them. A newer method called SuperSim (Turney 2013) does not require Web searching. These methods are evaluated on a dataset of 374 SAT questions that Turney and his collaborators have collected.

Many of the best results on this evaluation have been achieved by Turney in his own work. One interesting system not by Turney is BagPack (Herdağdelen and Baroni 2009), which could learn about analogies either from unstructured text or from ConceptNet 4.

Solving analogies over word embeddings is often described as comparing the difference $b_2 - a_2$ to $b_1 - a_1$ (Mikolov et al. 2013), but for the task of filling in the best pair for a_2 and b_2, it helps to take advantage of more of the structure of the question to provide more constraint than this single comparison.

In a sensible analogy, the words on the right side of the analogy will be related in some way to the words on the left side, so we should aim for some amount of relatedness between a_1 and a_2, and between b_1 and b_2, regardless of what the other terms are. Also, in many cases, a satisfying analogy will still make sense when it is transposed to "a_1 is to a_2 as b_1 is to b_2". The analogy "fire : hot :: ice : cold", for example, can be transposed to "fire : ice :: hot : cold". Recognizing this structure helps in picking the best answer to difficult analogy questions.

This gives us three components that we can weigh to evaluate whether a pair (a_2, b_2) completes an analogy: their separate similarity to a_1 and b_1, the dot product of differences between the pairs, and the dot product of differences between the transposed pairs. The total weight does not matter, so we can put these together into a vector equation with two free parameters:

$$s = a_1 \cdot a_2 + b_1 \cdot b_2 + w_1(b_2 - a_2) \cdot (b_1 - a_1) + w_2(b_2 - b_1) \cdot (a_2 - a_1)$$

The appropriate values of w_1 and w_2 depend on the nature of the relationships in the analogy questions, and also on how these relationships appear in the vector space. We optimize these parameters separately for each system we test, using grid search over a number of possible values so that each system can achieve its best performance. The grid search is performed on odd-numbered questions, holding out the even-numbered questions as a test set.

The weights found for ConceptNet Numberbatch 16.09 were $w_1 = 0.2$ and $w_2 = 0.6$. This indicates, surprisingly, that the comparisons being made by the transposed form of the analogy were often more important than the directly stated form of the analogy for choosing the best answer pair.

An Evaluation of Common-Sense Stories

The Story Cloze Test (Mostafazadeh et al. 2016) is a recent evaluation of semantic understanding that tests whether a

[4] It is sometimes important to distinguish *similarity* from *relatedness*. For example, the term "coffee" is related to "mug", but coffee is not *similar* to a mug. What a machine can learn from the connectivity of ConceptNet is focused on relatedness.

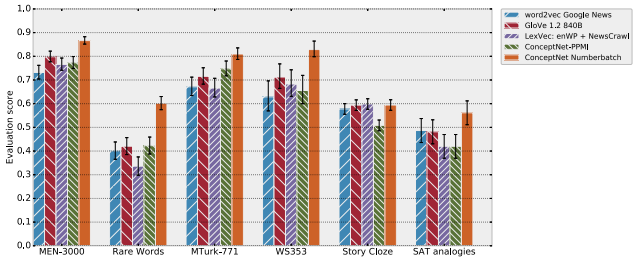

Figure 2: Performance of word embeddings across multiple evaluations. Error bars show 95% confidence intervals.

Evaluation	Dev	Test	Final
MEN-3000 (ρ)	.859	.866	.866
Rare Words (ρ)	.609	.586	.601
MTurk-771 (ρ)	—	.810	.810
WordSim-353 (ρ)	.828	—	.828
WordSim-353 Spanish (ρ)	.685	—	.685
Story Cloze Test (acc)	.604	.594	.594
SAT Analogies (acc)	.535	.588	.561

Table 1: The Spearman correlation (ρ) or accuracy (acc) of ConceptNet Numberbatch 16.09, our hybrid system, on data used in development and data held out for testing.

method can choose the sensible ending to a simple story. Prompts consist of four sentences that tell a story, and two choices are provided for a fifth sentence that concludes the story, only one of which makes sense.

This task is distinguished by being very challenging for computers but very easy for humans, because of the extent that it relies on implicit, common sense knowledge. Most systems that have been evaluated on the Story Cloze Test score only marginally above the random baseline of 50%, while human agreement is near 100%.

Our preliminary attempt to apply ConceptNet Numberbatch to the Story Cloze Test is to use a very simple "bag-of-vectors" model, by averaging the embeddings of the words in the sentence and choosing the ending whose average is closest. This allows us to compare directly to one of the original results presented by Mostafazadeh et al., in which a bag of vectors using GenSim's implementation of word2vec scores 53.9% on the test set.

This bag-of-vectors model uses no knowledge of how one event might sensibly follow from another, only which words are related in context. Improving the score of this model should not be portrayed as actual "story understanding", but it recognizes that sensible stories do not suddenly change topic.

Results and Discussion

Word Relatedness

Figure 2 compares the performance of the systems we compared across all evaluations. For word-relatedness evaluations, the Y-axis represents the Spearman correlation (ρ), using the Fisher transformation to compute a 95% confidence interval that assumes the given word pairs are sampled from an unobservable larger set (Bonett and Wright 2000). For the analogy and story evaluations, the Y-axis is simply the proportion of questions answered correctly, with 95% confidence intervals calculated using the binomial exact test.

The scores of our system on all these evaluations appear in Table 1, including a development/test breakdown that shows no apparent overfitting. The "Final" column is meant for comparisons to other papers and used in the graph. It uses the standard test set that other publications use, if it exists (which is the case for MEN-3000 and Story Cloze), or all of the data otherwise.

On all of the four word-relatedness evaluations, ConceptNet Numberbatch 16.09 (the complete system described in this paper) is state of the art, performing better than all other systems evaluated to an extent that exceeds the confidence interval of the choice of questions. Its high scores on both the Rare Words dataset and the crowd-sourced MEN-3000 and MTurk-771 datasets, exceeding the performance of other embeddings with high confidence, shows both the breadth and the depth of its understanding of words.

Analogy-solving system	Accuracy	95% conf.
BagPack (2009)	.441	.390 – .493
word2vec (2013)	.486	.436 – .537
SuperSim (2013)	.548	.496 – .599
LRA (2006)	.561	.510 – .612
ConceptNet Numberbatch	.561	.510 – .612

Table 2: The accuracy of different techniques for solving SAT analogies, including ConceptNet Numberbatch 16.09, our hybrid system.

SAT Analogies

ConceptNet Numberbatch performed the best among the word-embedding systems at SAT analogies, getting 56.1% of the questions correct (58.8% on the half that was held out for final testing). These analogy results outperform analogies based on other word embeddings, when evaluated in the same framework, as shown by Figure 2.

The analogy results also tie or slightly outperform the performance of best-in-class systems on this evaluation[5]. Table 2 compares our results to the other systems introduced in the "Solving SAT-Style Analogies" section: BagPack (Herdağdelen and Baroni 2009), the previous use of ConceptNet on this evaluation; LRA (Turney 2006), the system whose record has stood for a decade, which spends nine days searching the Web during its evaluation; and SuperSim (Turney 2013), the more recent system that held the record among self-contained systems. We also include the optimized results we found for word2vec (Mikolov et al. 2013), which scored best among other word-embedding systems on this task.

The results of three systems – SuperSim, LRA, and our ConceptNet Numberbatch – are all within each other's 95% confidence intervals, indicating that the ranking of the re-

[5]See http://www.aclweb.org/aclwiki/index.php?title=SAT_Analogy_Questions for a thorough list of results.

sults could easily change with a different selection of questions. Our score of 56.1% is also within the confidence interval of the performance of the average human college applicant on these questions, said to be 57.0% (Turney 2006).

We have shown that knowledge-informed word embeddings are up to the challenge of real SAT analogies; they perform the same as or slightly better than non-word-embedding systems on the same evaluation, when other word embeddings perform worse. In practice, recent word embeddings have instead been evaluated on simpler, synthetic analogy data sets (Mikolov et al. 2013), and have not usually been compared to existing non-embedding-based methods of solving analogies.

We achieve this performance even though the system, like other systems that form analogies from word embeddings, is only adding and subtracting values that measure the relatedness of terms; it uses no particular representation of what the relationships between these terms actually are. There is likely a way to take ConceptNet's relation labels into account and perform even better at analogies.

Story Cloze Test

The performance of our system on the Story Cloze Test was acceptable but unremarkable. ConceptNet Numberbatch chose the correct ending 59.4% of the time, which is in fact slightly better than any results reported by Mostafazadeh et al. (2016), including neural nets trained on the task. However, we could also achieve a similar score by using the same bag-of-vectors approach on other word embeddings. The best score of 59.9% was achieved by LexVec, with ConceptNet Numberbatch, GloVe, and word2vec all within its confidence interval.

This result should perhaps be comforting to those who aim to improve the computational understanding of stories. A bag-of-vectors approach may be marginally more successful at choosing the correct ending to a story than other approaches, but the performance of this approach has likely reached a plateau. It seems that any sufficiently high-quality word embeddings can choose the correct ending about 59% of the time, based on nothing but the assumption that the end of a story should be similar to the rest of it. Consider this a baseline: any representation designed to usefully represent the events in stories should get more than 59% correct.

Conclusion

We have compared word embeddings that represent only distributional semantics (word2vec, GloVe, and LexVec), word embeddings that represent only relational knowledge (ConceptNet PPMI), and the combination of the two (ConceptNet Numberbatch), and we have shown that the whole is more than the sum of its parts.

ConceptNet continues to be important in a field that has come to focus on word embeddings, because word embeddings can benefit from what ConceptNet knows. ConceptNet can make word embeddings more robust and more correlated with human judgments, as shown by the state-of-the-art results that ConceptNet Numberbatch achieves at matching human annotators on multiple evaluations.

Any technique built on word embeddings should consider including a source of relational knowledge, or starting from a pre-trained set of word embeddings that has taken relational knowledge into account. One of the many goals of ConceptNet is to provide this knowledge in a convenient form that can be applied across many domains and many languages.

Availability of the Code and Data

The code and documentation of ConceptNet 5.5 can be found on GitHub at https://github.com/commonsense/conceptnet5, and the knowledge graph can be browsed at http://conceptnet.io. The full build process, as well as the evaluation graph, can be reproduced using the instructions included in the README file for using Snakemake, a build system for data science (Köster and Rahmann 2012), and optionally using Docker Compose to reproduce the system environment. The version of the repository as of the submission of this paper has been tagged as `aaai2017`.

The ConceptNet Numberbatch word embeddings that resulted from this build process in September 2016 are the ones evaluated in this paper; they can be downloaded as prebuilt embeddings from https://github.com/commonsense/conceptnet-numberbatch, tag `16.09`.

Acknowledgments

We would like to thank the tens of thousands of volunteers who provided the crowd-sourced knowledge that makes ConceptNet possible. This includes contributors to Open Mind Common Sense and its related projects, as well as contributors to Wikipedia and Wiktionary, who are improving the state of knowledge for humans and computers alike.

References

Anacleto, J.; Lieberman, H.; Tsutsumi, M.; Neris, V.; Carvalho, A.; Espinosa, J.; Godoi, M.; and Zem-Mascarenhas, S. 2006. Can common sense uncover cultural differences in computer applications? In *Artificial intelligence in theory and practice*. Springer. 1–10.

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

Bond, F., and Foster, R. 2013. Linking and Extending an Open Multilingual Wordnet. In *51st Annual Meeting of the Association for Computational Linguistics: ACL-2013*, 1352–1362.

Bonett, D. G., and Wright, T. A. 2000. Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika* 65(1):23–28.

Breen, J. 2004. JMDict: a Japanese-multilingual dictionary. In *Proceedings of the Workshop on Multilingual Linguistic Resources*, 71–79. Association for Computational Linguistics.

Bruni, E.; Tran, N.-K.; and Baroni, M. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res. (JAIR)* 49:1–47.

Faruqui, M.; Dodge, J.; Jauhar, S. K.; Dyer, C.; Hovy, E.; and Smith, N. A. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, 406–414. ACM.

Halawi, G.; Dror, G.; Gabrilovich, E.; and Koren, Y. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1406–1414. ACM.

Hassan, S., and Mihalcea, R. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*.

Herdağdelen, A., and Baroni, M. 2009. Bagpack: A general framework to represent semantic relations. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, GEMS '09, 33–40. Stroudsburg, PA, USA: Association for Computational Linguistics.

Köster, J., and Rahmann, S. 2012. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics* 28(19):2520–2522.

Kuo, Y.-L.; Lee, J.-C.; Chiang, K.-Y.; Wang, R.; Shen, E.; Chan, C.-W.; and Hsu, J. Y.-J. 2009. Community-based game design: experiments on social games for commonsense data collection. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 15–22. ACM.

Lenat, D. B., and Guha, R. V. 1989. *Building large knowledge-based systems: representation and inference in the Cyc project*. Addison-Wesley Longman.

Levy, O.; Goldberg, Y.; and Dagan, I. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3:211–225.

Liu, H., and Singh, P. 2004. ConceptNet – a practical commonsense reasoning tool-kit. *BT Technology Journal* 22(4):211–226.

Luong, M.-T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. *CoNLL-2013* 104.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Miller, G.; Fellbaum, C.; Tengi, R.; Wakefield, P.; Langone, H.; and Haskell, B. 1998. *WordNet*. MIT Press Cambridge.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL: Human Language Technologies*, 839–849. San Diego, California: Association for Computational Linguistics.

Nakahara, K., and Yamada, S. 2011. Development and evaluation of a Web-based game for common-sense knowledge acquisition in Japan. *Unisys Technical Report* 30(4):295–305.

Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *AAAI*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)* 12:1532–1543.

Pustejovsky, J. 1991. The generative lexicon. *Computational linguistics* 17(4):409–441.

Salle, A.; Idiart, M.; and Villavicencio, A. 2016. Enhancing the LexVec distributed word representation model using positional contexts and external memory. *arXiv preprint arXiv:1606.01283*.

Singh, P. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. AAAI.

Singhal, A. 2012. Introducing the knowledge graph: things, not strings. *Official Google blog*. Retrieved from https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html on Dec. 1, 2016.

Speer, R., and Chin, J. 2016. An ensemble method to produce high-quality word embeddings. *arXiv preprint arXiv:1604.01692*.

Speer, R., and Havasi, C. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*. Springer. 161–176.

Speer, R.; Chin, J.; Lin, A.; Nathan, L.; and Jewett, S. 2016. wordfreq: v1.5.1. DOI 10.5281/zenodo.61937.

Turney, P. D. 2006. Similarity of semantic relations. *Computational Linguistics* 32(3):379–416.

Turney, P. D. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. 1:353–366.

von Ahn, L.; Kedia, M.; and Blum, M. 2006. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 75–78. ACM.

Xiao, M., and Guo, Y. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*, 119–129.

Zhao, K.; Hassan, H.; and Auli, M. 2015. Learning translation models from monolingual continuous representations. In *Proceedings of NAACL*.