KnowYourNyms? A Game of Semantic Relationships

Ross Mechanic, Dean Fulgoni, Hannah Cutler, Sneha Rajana,
Zheyuan Liu, Bradley Jackson, Anne Cocos,
Chris Callison-Burch and **Marianna Apidianaki***
Computer and Information Science Department, University of Pennsylvania
* LIMSI, CNRS, Université Paris-Saclay, 91403 Orsay
{rmechanic,dfulgoni,hcutler,srajana,zheyuan,jacbrad
acocos,ccb,marapi}@seas.upenn.edu

Abstract

Semantic relation knowledge is crucial for natural language understanding. We introduce *KnowYourNyms?*, a web-based game for learning semantic relations. While providing users with an engaging experience, the application collects large amounts of data that can be used to improve semantic relation classifiers. The data also broadly informs us of how people perceive the relationships between words, providing useful insights for research in psychology and linguistics.

1 Introduction

Knowledge of semantic relationships can help numerous NLP tasks that need to infer meaning from text, such as text classification, content analysis and query answering. We apply the "games with a purpose" methodology (von Ahn and Dabbish, 2004) to the task of discovering semantic relationships between words. Our aim is to collect a large volume of accurately labeled lexical relationships through this type of crowdsourcing. Gamification offers several advantages compared to a fully automatic or manual relation identification process since it enables acquiring considerable amounts of high quality data at no cost.

We have created a simple game called *KnowYourNyms?* with the tag line *Keep your brain on its toes*. It asks players to list word for a prompt in a short amount of time. As the seconds tick down, they type as many answers as they can for prompts like "What are kinds of seafood?" or "What are the parts of a volcano?" or "What's the opposite of fat?". Table 1 shows the hyponyms, meronyms and antonyms that our players provided in response to these questions. Their answers are useful as training data for natural language understanding applications and may provide useful insights for psycholinguistics research.

hyponyms of seafood: fish (54), shrimp (53), lobster (38), crab (36), clams (24), salmon (17), oysters (12), scallops (12), shellfish (10), mussels (10), cod (7), tuna (7), tilapia (5), whale (4), trout (4), octopus (4), shark (4), squid (3), prawn (3), haddock (3), flounder (2), catfish (2), swordfish (2), eel (2), sushi (2), bass (2), calamari (2), mussles (2).
Words suggested once: muss-, pearls, suslhi, prawns, schrimp, seal, hadsoxk, crab", sepia, scampi, scalop, seaweed, dolphin, fi-, seaww-, snapper, s-, pr-, seabass, jellyfish, cra-, muscles, oy-, soup, sardine, mahi, herrin, mussells, tipica, tun-, lob-, sa-, osyter, crawdad, roe, swai-, cram-, pa-, caviar, seewee-, carp, oyste-, sw-, musse-.
meronyms of volcano: lava (32), rock (12), magma (10), mountain (9), crater (9), smoke (7), eruption (7), ash (6), fire (6), vent (4), heat (4), mouth (3), steam (2), danger (2), dust (2), volcano (2), cone (2), core (2), geodes (2).
Words suggested once: crust, energy, moutain, hot, village, sulfur, mount-, caldera, throat, pummice, gas, top, side, sill, stones, sparks, motlen, lawa, japan, opening, soil, head, earth, metal, op-, cliff, cond-, cr-, pl-, flow, pressure, spout, clay, pollution, sediment, rim
antonyms of fat: thin (15), skinny (13), slender (5), slim (4), small (4), tiny (3), fit (3), trim (2), lean (2).
Words suggested once: delgado, svelt, narrow, bare, attractive, anorexic, teeny, underweight, bulemic, in shape, under-, wispy, healthy, light, smal-, little

Table 1: Example relationships provided by *KnowYourNyms?* players (with frequency counts).

Go to www.know-your-nyms.com to play *KnowYourNyms?*.

2 Related Work

Several games with a purpose (GWAPs) have been developed for gathering linguistic annotations for building resources and training systems (Chamberlain et al., 2013). Lafourcade (2007) and Fort et al. (2014) developed games for defining semantic relations and dependency relations in French. Chamberlain et al. (2008) created *Phrase Detectives* to annotate and validate things like co-reference. Jurgens and Navigli (2014) recently proposed using video games to link Word-

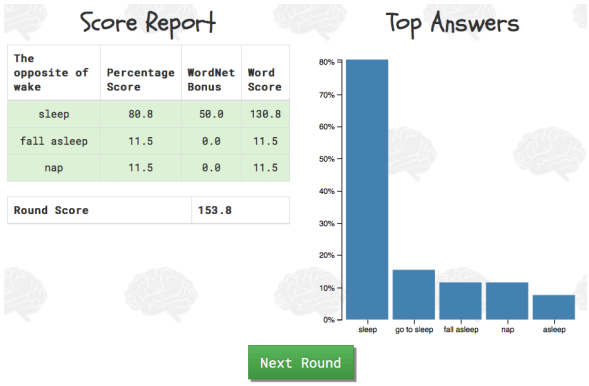

Figure 1: This example scoring page shows the scores for the player's words, and the top answers.

Net senses to images and perform word sense disambiguation. *KnowYourNyms?* gathers high quality semantic relationships between English words to increase the coverage of resources like WordNet and assign a taxonomic structure to the Paraphrase Database (Ganitkevitch et al., 2013). Additionally, it provides rich data for training relation detection systems like LexNET (Shwartz and Dagan, 2016), up to now trained on small training datasets (BLESS (Baroni and Lenci, 2011), EVALution (Santus et al., 2015), ROOT9 (Santus et al., 2016) and K&H+N (Necsulescu et al., 2015)).

3 System Overview

KnowYourNyms? is modeled after GWAPs like the ESP game or the Google Image Labeler, which use human-based computation to gather metadata to improve image recognition classifiers (von Ahn and Dabbish, 2004). At a high level, the application is simple. Once a user creates an account, she may start a round of the game. For each round, the system selects a specific word (called the "base word") and asks the user to name as many semantic relationship pairs for that word as possible in a set time limit. After the allotted time expires, these named pairs are recorded in our database and serve as data points for possible semantic relationships. The user then sees a display of her scoring performance, which is primarily based on how many other users named the same relationships for the given base word. In this way, the scoring is reminiscent of Family Feud, a popular game show that incentivizes answering questions in a way most similar to your peers. The scoring screen also shows the most popular answers to the question, in their appropriate distribution. Once completed, another round begins. The rounds are short (5-20 seconds, depending on the relation type), which makes the game fun and easy to play in short periods of time.

4 System Implementation

4.1 Architecture

The web application was built with the Django framework, using Python for all backend and database interaction and standard JavaScript, HTML, and CSS for the frontend, including the jQuery, d3.js, and Bootstrap JavaScript/CSS libraries. We used AWS Elastic Beanstalk, which deploys our Django web application to an AWS EC2 server. The application has multiple components that are important to the user experience, which are separated into three main views.

Welcome Screen This screen gives information about the purpose of the game, what semantic relationships are, how to play, and a little about our team. When a user is signed in this screen displays some statistics about the player, including number of completed rounds, total score, and average score per round. Four checkboxes are displayed, one for each playable semantic relationship type (synonyms, antonyms, hyponyms, and meronyms). These allow the user to select which relations to play. All are selected by default.

Game Play A timer begins immediately as the round starts. To answer the question prompt, the user may type as many semantic relationships as possible into text forms. Each discrete answer is known as an input word. Forms are dynamically generated upon pressing tab or enter, for however many input words are necessary during that round. At the end of 20 seconds, the round immediately ends and the user is directed to the scoring page.

Scoring Page Figure 1 shows what a player sees after the time elapses for a round. This scoring page displays two items to the player. The first is a table breakdown of all input words during the round, mapping each to a score for that word. It also includes the total round score. The second is a bar graph showing the top answers for that question. Here, users can observe which relations they identified or missed compared to the entire population.

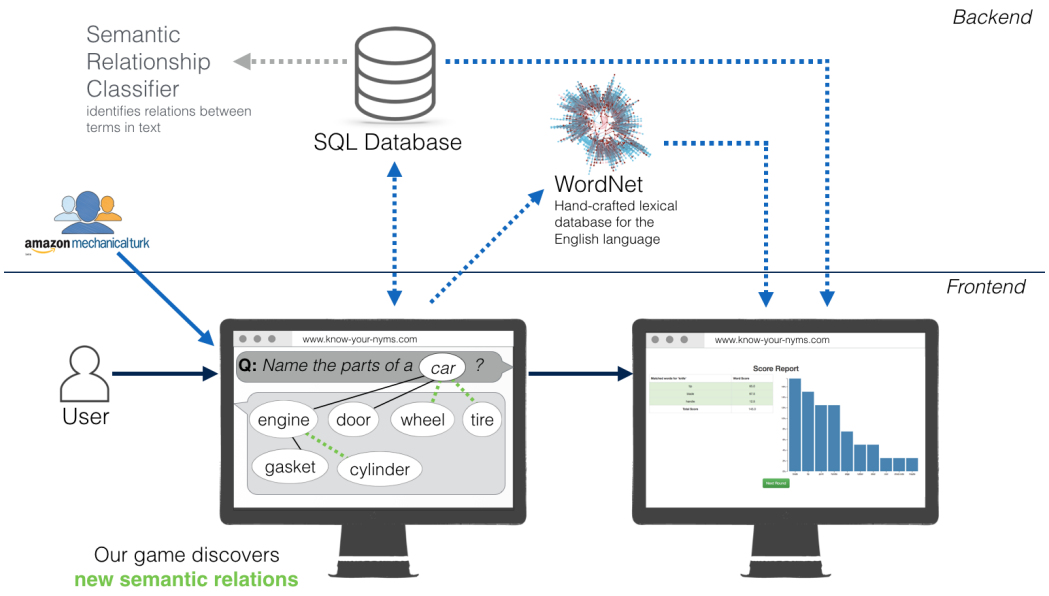

Figure 2: The application flow of the *KnowYourNyms?* game. The bottom half of the figure depicts the application functionality from the user's perspective (frontend). The top half of the figure shows the components of the system backend. Note that the "Semantic Relationship Classifier" is faded because we trained and tested the Classifier on the players' data in an offline setting (see Section 6.2).

4.2 Base Word Selection

Base words are those used for each round's question; they are the 'X' in a potential (X,Y) semantic relationship pair. Good base words are essential for good questions, as there aren't necessarily good synonyms for 'triceratops' or many parts of a 'sphere'. To address these issues, we build four separate vocabulary lists for the base words, one for each allowed semantic relationship type extracted from WordNet. We select base words that have at least one synonym or antonym, and at least three hyponyms or meronyms in WordNet. To make sure we don't ask users about rare words, which might discourage the users from continuing playing, we only retain unigrams and bigrams that occur at least 1,000,000 times in the Google n-grams corpus. Table 2 shows the number of base words retained from WordNet for each relation type. Finally, we have integrated a "skip" button which allows the users to skip queries for which they cannot think of any good relations.

Relation	Base Words
Synonyms	9,172
Antonyms	2,016
Hyponyms	4,107
Meronyms	678

Table 2: # of base words for each relation type.

4.3 Scoring

We incentivize players to generate many answers to each prompt by giving them a score at the end of each round. The score is based on the percentage of other users who named a word when they were given the same base word and relationship type. Finally, the score is also potentially augmented by a WordNet bonus, which is a simple boolean check of whether the word pair is linked by this specific relation in WordNet. The total score for each word is the sum of these values, sorted in descending order in the final score table.

4.4 Data Visualization

In order for users to see the most common responses for each round, a bar graph is included on the scoring page that shows the top 5 responses and the percentage of previous users who gave them. The percentages for scoring are calculated on the backend. On the frontend, we use the data visualization library d3.js, in order to dynamically create a bar graph that is scaled to the appropriate size for the window. This allows the graph to be seen on mobile devices, or to be dynamically resized as the user changes the size of a desktop window.

5 Design Decisions

5.1 User Identification

We require users to create an account. This design decision was mainly driven by quality control concerns. Since we don't expect all users to provide good answers, it is important that we be able to filter out malicious users, so that we can gather data that has sufficiently high quality for research purposes. An additional benefit of user identification is that it allows to not present a user with the same query several times, since this could skew the data.

5.2 Vocabulary Selection

The list of base words is traversed in a specific order by each user. Compared to fully random selection, this has the advantage of not repeating words until all have been played by the user. Presenting the user with the same words a few rounds apart is unacceptable from a user experience standpoint. Furthermore, having different users play the same words is important since it leads to better scoring and percentage visualization. Finally, this traversal is beneficial for learning high confidence relationships, as we collect data on fewer base words in a more concentrated way. To cover more words, we decided to allow a small amount of randomness which consists in drawing a word randomly from the whole vocabulary list every five items.

6 Evaluation

6.1 Crowdsourced Approach

To evaluate our game, we asked 160 crowdworkers to play *KnowYourNyms?* on Amazon Mechanical Turk for ten rounds each. Our intention was to seed the game with data so that normal users would receive scores based on words suggested by previous players. Although these workers were only asked to play ten rounds, many went on to play thirty, forty, or even a hundred rounds of the game. From these workers, we received over 15,000 user inputs. Table 3 gives a breakdown of the relations that we have collected so far. Here are some examples of the most frequently suggested word pairs for our relation types. Synonyms include **pony-horse**, woods-forest, **woods-trees**, **marching-walking**, **electricity-power**, **four-quad**, **looking-seeing**, **frequent-often**, **woody-forest**, and **pester-annoy**. Antonyms include **sleep-awake**, limited-unlimited, prefix-suffix, desirable-undesirable, **similarity-difference**, **similarity-different**, hitch-unhitch, immature-mature, wake-sleep, and **sterile-dirty**. Meronyms include **knife-handle**, knife-blade, chain-link, **woods-trees**, book-cover, **writings-words**, **ice-water**, **month-days**, **aquarium-fish**, and **chain-metal**. Hyponyms include **seafood-fish**, seafood-shrimp, **seafood-lobster**, **sleep-deep**, **similarity-same**, **seafood-crab**, **plaster-paris**, Asian-Chinese, Asian-Japanese, and **hitch-trailer** Bold items are relations that are **not present** in WordNet.

Users (n)	Rels	Rels not in WordNet	Rels in WordNet
all	17,603	16,813	790
n<=3	15,895	15,265 (96%)	630
3<n<=5	724	672 (93%)	52
5<n<=15	794	723 (91%)	71
15<n<=30	153	126 (82%)	27
n>30	37	27 (73%)	10

Table 3: The number of relations (rels) learned at different confidence levels, where confidence is measured by the number of users (n) who named the relation. We compare this to the number of relations found in WordNet for the same base words.

We surveyed the crowd workers about their feelings about the game and whether or not they would play again. The first 30 crowd workers played the game before anyone else had played, so many of their scores were empty (the game relies on previous players). Those workers rated the game on average 3.9/5 on experience and 3.8/5 on likelihood of playing again. However, our second group of crowd workers was given the game with many more of the rounds already played, which improved scoring. These workers rated the experience 4.46/5 on average, and 4.43/5 for likelihood of playing again. Moreover, many of the second round of workers left comments stating that they "loved this addicting game", that the game "is fun", "makes you think fast" and "really wakes up the brain", and made useful suggestions for improvement. The positive reaction about playing the game (especially the shift in positivity as the scoring became more clear) is evidence that this game may work on a larger scale, and may allow us to gather important word relationship data from players for free.

6.2 Classifier Evaluation

To demonstrate how this game could be used to collect training data for semantic relation classi-

	Count Train/ Val	Count Test	P	R	F
meronyms	1162	248	0.44	0.91	0.59
hyponyms	337	313	0.50	0.01	0.01
antonyms	1279	22	0.25	0.77	0.38
synonyms	859	14	0.02	0.14	0.03
random	1038	354	0.58	0.40	0.47
total / avg	4675	951	0.50	0.41	0.34

Table 4: **P**recision, **R**ecall, and **F**-Score of the LexNET semantic relation classifier, when trained and evaluated on data collected by *KnowYourNyms?*.

fiers, we used our players' data to train and evaluate a state-of-the-art semantic relationship classifier, LexNET (Shwartz and Dagan, 2016). Our dataset consisted of 8613 meronym, antonym, hyponym and synonym pairs proposed by at least five users, and 6228 random word pairs. From these 14,841 pairs, we extracted a subset of 951 pairs for testing and used the remaining 4675 pairs whose constituent words did not overlap with the test set for training and validation. The classifier achieved an overall weighted average F-Score of 0.34 over the test set. The full results of this experiment are given in Table 4.

7 Discussion

One of the challenging aspects of making this game fun to play is selecting words and relation types that are easy for people to think of answers for. Despite our attempts to filter the vocabulary sets drawn from WordNet to be high frequent words with several WordNet relations, we found that many players were stumped by some of our questions. Here are examples of the questions that most users pressed the "Pass" button for:

- What are kinds of geology? (71% passed)
- What are kinds of a saver? (70%)
- What is the opposite of conception? (67%)
- What is the opposite of differentiated? (67%)
- What are kinds of hormones? (67%)
- What is another word for notorious? (60%)
- What are kinds of sinking? (56%)
- What are kinds of barley? (56%)

Some prompts are clearly more difficult for users to answer than others. We hypothesized that abstract words (e.g. *geology*, *sinking*, *dissolution*) are more difficult to provide relations for than concrete words. An indicator of annotation difficulty for a word is the number of times users choose to skip it: if they cannot think of any good relationships, users can choose to pass to the next round. We calculate the correlation between word difficulty – measured as the ratio of the number of times the word was skipped to the number of times it was seen – and concreteness scores in the dataset built by Brysbaert et al. (2014) (hereafter CONCRETE) which contains ratings for 37,058 English words and 2,896 two-word expressions. Words are ranked on a 5-point rating scale going from abstract words (low values) to words with concrete meaning (high values). We expect abstract words to be more difficult to handle and more frequently skipped by our users compared to concrete words.

We perform the correlation calculation on 412 lemma-relation pairs extracted from *KnowYourNyms?*. From these, 40 correspond to specific terms and named entities (e.g. *cytochrome, methyl, Utah, Mexico, etiology, flora, Maryland*) that are not in CONCRETE (it only includes words known to 85% of the annotators and excludes proper names). We intend to use existence in CONCRETE as a criterion for identifying words that would be too difficult for the annotators and should be excluded from our game.

The Pearson correlation results for the remaining 372 words indicate a negative correlation of -0.2007 between word difficulty and concreteness ($p < 0.001$), confirming our assumption that more abstract words are more difficult to handle. Correlation for the 99 lemmas in CONCRETE that were seen at least 10 times by our crowdworkers is even higher, - 0.3851 ($p < 0.001$),

Finally, we intend to analyze the collected relations in the light of typicality and gradual semantic category membership, as proposed in (Vulić et al., 2016), to make them more useful for textual entailment tasks.

8 Conclusions and Future Work

KnowYourNyms? gamifies the process of gathering pairs of words holding specific semantic relationships that are not found in existing resources. While providing users with an entertaining experience, our application enables collection of large amounts of data that can be used to improve se-

mantic relation classifiers and content analysis tools. This application offers exciting possibilities for further development. As the number of players grow, our lexical relation dataset will keep expanding. This will provide new opportunities for evaluation in full-blown applications and will richen our understanding of how people perceive word relations.

9 Software and Data

We release the software that underlies our game under the BSD open source license. We provide instructions on how to set up your own instance of the game and populate it with your own base words and semantic relationship types. The software is available at https://github.com/rossmechanic/know_your_nyms/. A file containing the semantic relations collected during our initial testing of the game is also included in the repository.

Acknowledgments

We would like to thank Ani Nenkova and Jonathan Smith for the discussions and useful feedback on this project, and the Mechanical Turk workers who did the play testing.

This material is based in part on research sponsored by DARPA under grant number FA8750-13-2-0017 (the DEFT program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government. This work has also been supported by the French National Research Agency under project ANR-16-CE33-0013.

References

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of GEMS*. Edinburgh, UK, pages 1–10.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods* 46(3):904–911.

Jon Chamberlain, Karën Fort, Udo Kruschwitz, Mathieu Lafourcade, and Massimo Poesio. 2013. Using games to create language resources: Successes and limitations of the approach. In *The Peoples Web Meets NLP*, Springer, pages 3–44.

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2008. Phrase detectives: A web-based collaborative annotation game. In *Proceedings of I-Semantics 08*. pages 42–49.

Karën Fort, Bruno Guillaume, and Hadrien Chastant. 2014. Creating Zombilingo, a Game With A Purpose for dependency syntax annotation. In *Proceedings of GamifIR '14*. Amsterdam, The Netherlands, pages 2–6.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL/HLT*. Atlanta, Georgia, pages 758–764.

David Jurgens and Roberto Navigli. 2014. It's All Fun and Games until Someone Annotates: Video Games with a Purpose for Linguistic Annotation. *TACL* 2:449–464.

Mathieu Lafourcade. 2007. Making people play for Lexical Acquisition with the JeuxDeMots prototype. In *Proceedings of SNLP'07*, Pattaya, Thailand.

Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading Between the Lines: Overcoming Data Sparsity for Accurate Classification of Lexical Relationships. In *Proceedings of *SEM*. Denver, Colorado, pages 182–192.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine features in a random forest to learn taxonomical semantic relations. In *Proceedings of LREC*. Portoroz, Slovenia.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. In *Proceedings of Linked Data in Linguistics: Resources and Applications*. Beijing, China, pages 64–69.

Vered Shwartz and Ido Dagan. 2016. CogALex-V Shared Task: LexNET - Integrated Path-based and Distributional Method for the Identification of Semantic Relations. In *Proceedings of CogALex-V*. Osaka, Japan, pages 80–85.

Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vienna, Austria, pages 319–326.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2016. HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment. *CoRR* abs/1608.02117.