A Game with a Purpose for Building Crowdsourced Semantic Relations Datasets for Named Entities

André Fernandes dos Santos(✉) and José Paulo Leal

CRACS & INESC Tec LA, Faculty of Sciences, University of Porto, Porto, Portugal
afs@inesctec.pt, zp@dcc.fc.up.pt

Abstract. Semantic measures evaluate and compare the strength of relations between entities. To assess their accuracy, semantic measures are compared against human-generated gold standards. Existing semantic gold standards are mainly focused on concepts. Nevertheless, semantic measures are frequently applied both to concepts and instances. Games with a purpose are used to offload to humans computational or data collection needs, improving results by using entertainment as motivation for higher engagement. We present Grettir, a system which allows the creation of crowdsourced semantic relations datasets for named entities through a game with a purpose where participants are asked to compare pairs of entities. We describe the system architecture, the algorithms and implementation decisions, the first implemented instance – dedicated to the comparison of music artists – and the results obtained.

Keywords: Semantic Relations · Crowdsourcing · Dataset · Gamification

1 Introduction

Semantic measures (SMs) evaluate how close or how much related are the meanings of two *things* (e.g. concepts, words, sentences or named entities). Due to the inherently psychological nature of this process, SMs are evaluated using datasets that average the human perception of those semantic relationships.

There are several such datasets available. These are usually built by asking participants to rate relationships between pairs of things. Requesting a numeric value makes it easier to be used by computers. People, however, have an easier time comparing things than assigning numeric values.

Most of these datasets are focused on the comparison of concepts, with very few concerning named entities. Semantic measures, however, are frequently applied also to named entities.

To address both shortcomings of the current methods for building semantic datasets, we present Grettir, a platform for creating crowdsourced gold standards for semantic measures between named entities. Grettir can be used to implement games in which users are asked to pick the most related pair of named entities

K. Arai (Ed.): SAI 2023, LNNS 739, pp. 422–439, 2023.
https://doi.org/10.1007/978-3-031-37963-5_30

among a set of three. The players choices are used to generate a list of pairs of entities sorted by the strength of their semantic relation. *Shooting Stars* is the first instance of Grettir (and currently, the only one). The theme for this game is music artists. Players are presented three music artists and are asked to find which artist is less related to the other two. Figure 1 presents the main view of Shooting Stars.

Fig. 1. Shooting Stars Main View.

The contributions described in this paper can be summarized as follows:

1. We propose using games to build semantic relations datasets to increase the motivation of the users.
2. We created a framework for generating such games in which player's activity builds a semantic relations dataset.
3. We generated a first instance dedicated to musical artists.
4. We obtained a small dataset consisting of a list of pairs of music artists sorted by the strength of their relationship.
5. We performed usability and validation tests.

This paper is structured as follows. Section 2 provides a brief overview on the field of semantic measures, their evaluation using semantic datasets and how gamification has been used to build crowdsourced human-generated datasets. Section 3 describes our approach for implementing Grettir. Section 4 presents its

architecture, and Sect. 5 details the implementation and the main algorithms. Sections 6 and 7 present, respectively, the methodology followed for testing and validating the first instance of Grettir, and the results obtained in those tests, which are then discussed on Sect. 8. Section 9 summarizes the research described in this paper, lists some future improvements and highlights its main contributions.

2 Background

2.1 Semantic Measures

Semantic measures (SM) are a form of evaluating how close or how much related are the meanings of two *things*. SMs are widely used to identify the nature and strength of the semantic relationships between concepts, words, sentences or named entities, among other elements. This evaluation is useful in areas such as computational linguistics, language understanding, bioinformatics and information retrieval.

Semantic measures are based on the analysis of information describing elements extracted from semantic sources. These semantic sources can be classified as either *unstructured* (e.g. plain text), *semi-structured* (e.g. dictionaries), or *structured*. The latter include a large range of computer understandable resources, from structured vocabulary to highly formal knowledge representations.

Semantic similarity takes into account only taxonomic relationships. Semantic relatedness considers all types of relationships. While *car* and *train* are similar concepts because both of them are types of *vehicles*, *car* and *wheel* are related concepts because the former is part of the latter [13].

The *knowledge-based approach* to computing semantic measures relies on semantic graphs extracted from structured sources. The properties of a semantic graph, of its nodes and edges contain semantic evidence regarding the interconnections and the semantics of relationships between elements. This information is analyzed to produce a (usually numeric) value. Within this approach, several methods have been defined to compare elements in single and multiple knowledge bases: information theoretical methods [1,8], feature-based methods [9,10] and structural methods. These use the graph structure (nodes and arcs) to compare elements, relying on graph-traversal approaches, such as shortest path or random walk techniques.

Several publicly available semantic graphs are currently composed of billions of nodes and edges – DBpedia, Freebase, OpenCyc, Wikidata and YAGO have been compared by [4]. Such graphs frequently include not only information regarding concepts and their relations, but also instances. Several path-based semantic measures applied to such semantic graphs can be used to compare concepts with concepts, but also concepts with instances or instances with instances [3,7].

2.2 Evaluation of Semantic Measures

Accuracy of a measurement can be defined as *"closeness of that measurement to the real value being measured"* [2]. When discussing SMs, however, the notion of "real value" is far from straightforward: it only has meaning when compared against values of the same measure for other pairs of entities (i.e. to have an arbitrary measure reporting *house* and *building* as having a similarity of 1 is meaningless on its own), and it represents an inherently subjective appreciation.

Consequently, there are two methods for evaluating the accuracy of SMs: *directly*, through comparison with averaged values reported by humans, or *indirectly*, by measuring the performance of applications which are highly dependent on the semantic measures (e.g. term disambiguation, classification) [35]. When evaluating SMs directly, gold standard datasets are often gathered by recruiting a variable number of people and asking them to rate (i.e. to assign a numeric value to) the semantic relationship between pairs of entities. In the more recent years, crowdsourced marketplaces such as Amazon Mechanical Turk have been leveraged to build datasets by providing online workers with a small monetary reward to complete the same task [5,11,12].

Asking people to rate relationships presents several challenges [15]:

1. It requires assigning a numeric value to a subjective appreciation.
2. It requires remembering the values attributed to previous pairs, so that the current rating can be contextualized.
3. It should also require knowledge of the next pairs, for the same reason.
4. The necessary precision is not known beforehand.

Instead of requiring participants to assign a numeric value to semantic relations, when building the MEN dataset, [6] asked them to choose the most related pair among two candidate pairs. According to the authors it would *"constitute a more natural way to evaluate the target pairs, since humans are comparative in nature"*, among other operational reasons.

In the process of direct evaluation, SMs are applied to the pairs contained in a gold standard dataset. Then, the correlation between the resulting values and the ones in the dataset is calculated. Measuring the correlation allows to focus on covariance rather then the absolute values. Pearson's correlation coefficient [20], a measure of the linear correlation between two sets of values, is one of the most common ways of evaluating how similar are the results of a SM and a semantic dataset.

Some measures and datasets do not provide a numeric value for the semantic relationships between entities, but instead give a list of pairs, sorted by the strength of their relationship – e.g $f(house, building) > f(house, phone) > f(phone, steak)$. In such cases, where only the order (or rank) is relevant, it is common to use Spearman's rank correlation coefficient [19] instead.

Semantic gold standards play an important role in the direct evaluation of semantic measures. The quality of the datasets themselves, on the other hand, is often considered to be the inter-agreement of participants – the correlation

between the scores given by human annotators – which is often calculated using the same approaches: Pearson's or Spearman's correlation coefficients.

Despite the importance of benchmarks in the direct evaluation of semantic measures, and the previously mentioned fact that semantic measures are often applied to instances, most available benchmarks are focused on concepts. Harispe et al., for example, provide a list of more than 20 existing semantic measures datasets [35], but it reports only one [3] as including entities.

2.3 Games Beyond Entertainment

Serious games are games whose primary purpose is not entertainment [14]. These include games for training and simulation, education, health, tourism, among others [16]. Serious games can be classified according to their gameplay, purpose and scope under the G/P/S model [14]. Casual games is a genre with a loose and somewhat controversial definition [33], but typically includes games with simple controls, easy-to-learn gameplay and support for short play sessions [31]. It includes older games such as the Solitaire card game from Windows 3.0, and Tetris [32]. More recently, casual games have become increasingly popular in mobile devices [34].

Games with a purpose (GWAPs) are games in which computational processes are outsourced to humans in an entertaining way [17,18]. This means that players are expected to play the game for fun, but, aware or not, while they do they are also contributing to some other goal. GWAPs have been used for music and sound annotation [21] and metadata validation [22], corpora annotation [23], sentiment analysis [24], and more, along with tasks related to linked data and the semantic web [25–27]. Siorpaes and Hepp, notably, have done extensive research on using GWAPs to help weave the Semantic Web [28,29]. More recently, WordGuess is a GWAP for vocabulary training [30].

3 Approach

Grettir is a platform to build semantic relationship datasets for named entities. These datasets can be used as gold standards in the evaluation of semantic measures. To improve the process of enlisting human participants, the system is implemented as a casual game in which participants compete asynchronously with each other while their answers are used to build the dataset.

Players are asked to pick the odd element (*the intruder*) in a group of three elements. Implicitly, they are choosing, from a set of three entities $\{A, B, C\}$, which pair is the most strongly related. For example, if the player picks the entity A as the intruder, they are simultaneously asserting the elements of the pair $\{B, C\}$ as being more related to each other than the elements of both $\{A, B\}$ and $\{A, C\}$.

This allows sorting the pairs of entities without explicitly asking the players to assign a numeric value to the strength of the relationships.

3.1 Gameplay

Each game in Grettir is composed of 10 rounds. In a game, three players play against each other: a single human player and two automated bots – artificial intelligence-based players, whose names are randomly picked from robot characters from popular culture (e.g. 2001: A Space Odyssey's HAL, Futurama's Bender or Portal's GLaDOS).

In each round, three named entities are displayed. The human player is asked to *find the intruder* among them, i.e. to pick which of the three entities is less related to the other two. Figure 1 presents the game screen for Shooting Stars (described in Sect. 6.2), displaying the names and pictures of three musical artists.

The human player makes their pick, and it is compared against the picks of the bots (described in Sect. 5.1). The *correct* pick[1] is considered to be the one shared by at least two of the players (the user and the two bots). Every player who picked that entity is awarded 10 points. The player who picked differently is awarded no points. If every player picks a different entity, there is no correct answer, and no one wins any points.

When all the rounds for that game have been played, the human player is presented with the score board. The points won during this game sequence are added to their total score, and they can proceed to play another (different) game sequence. A diagram representing the life-cycle of a Grettir game can be found in Fig. 2.

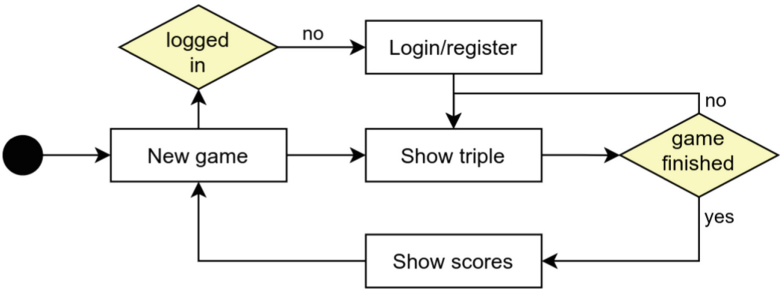


Fig. 2. Game Play Activity Diagram.

3.2 Game Design Elements

Grettir was implemented as a game to try to maximize user engagement. Arguably a higher level of engagement leads to more players and more games played, and as a result, better and larger datasets obtained. To this end we added several common game design elements to the system:

[1] Being a subjective measure, there is no right or wrong answer when evaluating relatedness. But the game format demands winners and losers. Grettir uses the most picked entity to make that decision.

Opponents: In each game, a single human player is faced by two opponents, which are in fact automated bots whose game picks follow the distribution of past players picks (see Sect. 5.1).

Goal: The objective of the player is to find which of the three entities displayed is less like the other two. Because the bots play according to past players choices, the goal is actually to guess what the common opinion regarding the three entities is.

Points: Each player who picks the correct entity in a round is given 10 points. No points are awarded if every player picks a different entity. No penalties are given for choosing incorrectly.

Leaderboards: During the course of a game the points for each round are added and displayed to the user. At the end of the game, the leaderboard for the game is presented. Additionally, the global leaderboard is also displayed, presenting the scores for the totality of the games played by the top ranking players.

Adjusted difficulty: Each time a player starts a new game, the sequence of triples of entities for that game is generated taking into account the current estimated difficulty of the triples, and the past performance of the player (see Sect. 5.1). Stronger players are given harder triples, while weaker players are presented easier triples.

Games implemented with Grettir can be classified as casual serious games with a purpose. They present several features which make them easier to play:

Mobile ready: Grettir is implemented with a HTTP API. This allows the front-end of each instance to be implemented as a web application, a desktop application or a mobile native application (or anything that can speak HTTP).

No registration needed: When users land on the main page they can start playing immediately, no previous registration needed.

Resumable on other devices: Creating an account allows the player to sign in and resume an ongoing game on another device.

4 System Architecture

Grettir has been implemented as a modular application. It is composed of three main components: the backend API, the game frontend and the statistics backoffice. A representation of the Grettir architecture can be found in Fig. 3.

The game frontend is the part of the application that is visible to the human player. It can be written in anything that can make HTTP requests, as it must fetch from, and write data to, the backend API. This component should be customized for each instance of Grettir to ensure that it matches the game theme. For Shooting Stars (presented in Sect. 6.2), it consists of a single page responsive web application developed in Vue.js. Players are allowed to play without registering or logging in. However, if they do, they are able to resume the game on other devices. When registering, players can optionally fill in some additional

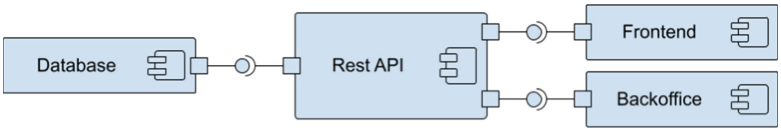


Fig. 3. Grettir Architecture.

personal information, such as age and gender, which can later be used to perform a more detailed analysis of how these attributes might influence the perception of relatedness.

The stats backoffice is meant to be used by the game administrators. It provides quick access to multiple charts which give a real-time overview of the system status: for example, the number of players registered each month, the average number of games played by each player in each month, or the distribution of players by gender. Additionally, it also displays the list of players, and the list of pairs sorted so far. This backoffice uses Charts.js and it also relies on the backend API.

The third component is the main engine of the game. The backend API is a Node.js application which exposes a MongoDB database by making available a RESTful interface in JSON format. The database is used to store the players profiles, the entities used in the game, and all the data needed to operate the game and generate new game sequences (e.g. games played and players picks). An entity relationship diagram for the database can be found in Fig. 4. The API handles all the requests coming from the game frontend and the statistics backoffice. It is also responsible for making all the calculations needed to generate new game sequences for players, and to extract the list of pairs sorted by their relatedness value, as described in Sect. 5.1).

5 Implementation Decisions

The goal of Grettir is the production of human-generated gold standards for semantic relationships. These are outputted as a list of pairs of entities, sorted from the most related pair to the least related one. To build such lists, players are presented with triples of entities, and must choose which entity is less related to the other two. Implicitly, they are choosing the most related pair (i.e. the other two entities). The data gathered from all players choices must then be processed to allow the sorting of the pairs.

Each time a player starts a new game the system needs to select triples to generate a new game sequence. Prioritizing triples corresponding to pairs which are ambiguous (that is, whose results are tied) would ensure that the ambiguity would be resolved as soon as possible. However, constraints related to gameplay concerns must also be considered. For example, a player should not be asked to play the same triple twice, and the difficulty of the triples being played should match their skill level.

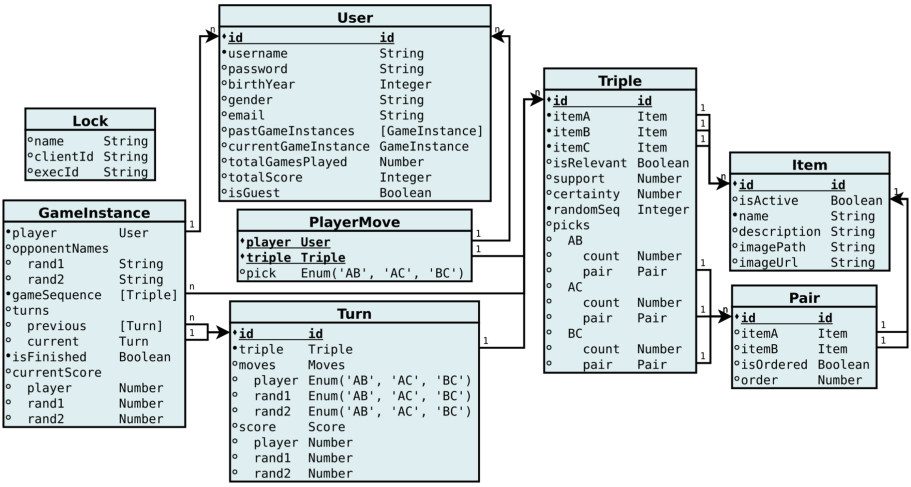


Fig. 4. Entity Relationship Diagram.

Given the uncertainty on the total number of games that will ever be played, games sequences are initially generated using a smaller subset of entities which is gradually extended when:

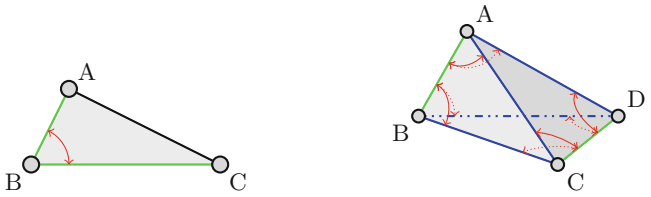
1. at least one player has played all the possible triples with the current set of entities; or
2. the players picks up to that point are sufficient to unambiguously generate a sorted list of pairs of entities.

This maximizes the size of the dataset and minimizes the ambiguity of the pairs, while still taking into account the players experience.

5.1 Algorithms

The main algorithms at the core of this game are the sorting of the pairs of entities and the generation of new game sequences for players. These two are interconnected: when generating a new game sequence, triples of entities are picked taking into account which pairs are still considered ambiguous; additionally, the pairs are sorted according to their relatedness using data gathered from the triples played.

Pair Comparison. For each possible triple of entities, three fields keep track of how many times each of the pairs has been chosen: `picks.AB.count`, `picks.AC.count` and `picks.BC.count`. These values are used to compare pairs. It can be graphically represented by the comparison of the sides of a triangle where A, B and C are the vertices, and the width of its sides is proportional to the corresponding `count` field (Fig. 5a).



(a) Direct (b) Indirect

Fig. 5. Pair Comparison Visual Representation.

Let C_R be a function that compares the strength of the relatedness R between the elements of two pairs. If the pairs share a common element, e.g. $\{A, B\}$ and $\{B, C\}$, ithen $C_R(R_{\{A,B\}}, R_{\{B,C\}})$ can be calculated directly by looking into the values of `picks.AB.count` and `picks.BC.count` of the triple $\{A, B, C\}$. The result is -1, 0 or 1 if A and B are, respectively, more, equally or less related than B and C.

$$C_R(R_{\{A,B\}}, R_{\{B,C\}}) = \begin{cases} -1 & \text{if } R_{\{A,B\}} < R_{\{B,C\}} \\ 1 & \text{if } R_{\{A,B\}} > R_{\{B,C\}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This comparison is transitive. If $\{A, B\}$ are more related than $\{\chi, \gamma\}$ and the latter are more related than $\{C, D\}$, then $\{A, B\}$ are more related than $\{C, D\}$:

$$R_{\{A,B\}} > R_{\{\chi,\gamma\}} \wedge R_{\{\chi,\gamma\}} > R_{\{C,D\}} \implies R_{\{A,B\}} > R_{\{C,D\}} \tag{2}$$

Pair comparison is more complex when the pairs do not share an element: $\{A, B\}$ and $\{C, D\}$. In such cases, there are four pairs directly comparable to the two original ones: $\{A, C\}$, $\{A, D\}$, $\{B, C\}$, and $\{B, D\}$. By applying the transitivity rule stated in Eq. 2 we can sum the results of comparing both of the original pairs with all the intermediate (and directly comparable) ones and obtain an indirect comparison (Eq. 3). We can extend the previous graphical representation to the comparison of two sides of a tetrahedron which do not share a vertex (Fig. 5b).

$$C_R(R_{\{A,B\}}, R_{\{C,D\}}) = \sum C_R(R_{\{A,B\}}, R_{\{\chi,\gamma\}}) + C_R(R_{\{\chi,\gamma\}}, R_{\{B,C\}}) \tag{3}$$

Sometimes pair comparison is inconclusive, either because there are not yet enough picks for those triples/pairs or because they are contradictory (imagine a regular triangle and tetrahedron on Fig. 5). In such cases, those pairs are considered, at that moment, *indistinguishable* from each other. This might mean that more data regarding those pairs is needed (i.e. more players playing the triples involved in their comparison) or these pairs might really not be consensual among the players.

Pair Sorting. This process of building the sorted list of pairs is performed recurrently, triggered by the necessity of determining which triples are the most relevant and should therefore be included in future game sequences (see next section on **Triple Selection** for more detail).

This process consists of creating an empty list and performing sorted inserts of pairs. In each insertion, the candidate pair is compared against some of the pairs already on the list (which ones depends on the insertion algorithm being used). Whenever a candidate pair P_C is found to be *indistinguishable* from a pair P_L already on the list (their comparison returns a 0), the process is halted. The triples corresponding to the comparison of P_C and P_L are marked as `isRelevant` (more on that later) and the list of pairs inserted so far is returned as the current sorted list.

In order to decrease the potential inconclusive comparisons (which halt the sorting process), we decided to use an AVL tree to make the ordered insertions, as it minimizes the number of comparisons needed. Furthermore, the order in which pairs are attempted to be inserted is given by the order of the last sorting of the pairs list or their `id` field.

Triple Selection. The reliability of the results for a `Triple` can be measured by observing two fields:

- **support:** a counter which is incremented every time the triple is played,
- **certainty:** the maximum value found in the picks `count` fields of the triple.

Low values of support are found in triples which have been played only a few times. For triples with high support, low values of certainty mean that players tend to disagree; high values in both fields are found in triples which, on average, are easy for players to disambiguate.

New game sequences are generated by choosing triples which are more useful for the pair sorting process. These correspond to pairs which halted the previous generation of the sorted list of pairs due to their ambiguity. This is the reason why such pairs are marked as `isRelevant` during the sorting process. Prioritizing these triples ensures the efficiency of the pair sorting algorithm in terms of the number player picks required.

On the other hand, there are concerns regarding gameplay and player experience. Users should enjoy the game because better engagement might translate into more games played or more shares with family and friends. To this end, players are never asked to play the same triple twice; every game sequence a user plays is made of triples that user has never seen before. Additionally, we use the players previous scores to infer their ability.

Taking all this into account, the selection of triples for a new game sequence for a player P is performed as follows:

1. First, we start by excluding all the triples already played by P.
2. Then we sort the remaining triples. Triples marked as `isRelevant` are placed at the top, and the others at the bottom.

3. This sorting is refined using the `certainty` field: a triple with low certainty is more in need of players input that one with high certainty.
4. This results in a list of triples which is arguably sorted from the most difficult (relevant triples, related to inconclusive pair comparisons, and with low certainty values) to the least ones (triples related to already sorted pairs, with high values of certainty).
5. Then we use the players ability to determine the place in the list from which to select triples (better players will have triples selected from the top, worse players, from the bottom).

Bot Picks. In each round, the automated bots must decide which entity to pick. They do so by generating a random pick following the distribution of past players picks. If for a given triple, entity A has been chosen by players 70% of the times, entity B was chosen 20% and entity C 10%, the bots choice will be a weighted random choice with those weights. This means that bots will tend to follow the opinion of the majority, while still allowing some variance.

6 Methodology

This section describes the usability and validation tests we performed for Grettir, and the creation and deployment of Shooting Stars, the first instance of Grettir.

6.1 Usability and Validation Tests

Before publicly deploying the first instance of Grettir, we performed tests of usability and user validation. The usability testing consisted of asking seven Msc-level computer science students to play test instances of Grettir, observing their behavior, and in the end asking them to fill out a form with usability-related questions. According to the literature, this number of users should be enough to identify most usability issues [36,37]. For the validation of our algorithms, we instantiated Grettir with items from publicly available semantic datasets, and compared the sorted lists of pairs returned by Grettir with those datasets. Both experiments were performed simultaneously: the users tested the usability of the platform while generating data for our validation assessment.

Test users were asked to fill out a form after playing these instances of the game. Most questions asked them to rate statements on a Likert scale, e.g. "I understood that I was playing against the average results of other players" or "The 'Show Tutorial' feature is helpful". The remaining questions were open ended questions devised to let users communicate what they felt were the biggest issues and points to be improved in the game, e.g. "Did you find anything wrong in the game's UI? (layout problems, bugs, etc.)".

For the validation of our algorithms, we needed lists of pairs of items small enough so that a few users could provide enough data to sort them. Additionally, given our sorting algorithms tendency to produce dense lists (where the pairs sorted include most combinations of items with each other), our gold standard

lists should be similar. For this purpose, we analyzed the semantic datasets listed in [35] which were available online, selecting the ones containing the largest *cliques*. Consider a semantic dataset D composed of tuples (c_1, c_2, R) where c_1 and c_2 are concepts and R is the strength of their relatedness. We then define a clique Q_D as a set of concepts belonging to D in which for any concepts q_1 and q_2 belonging to Q, there is a corresponding tuple (q_1, q_2, r) in D.

The resulting cliques were not big enough for our purposes (less than 8 items), so we extended them, increased their size by adding entities that, despite not being connected in D to all the other entities in S, were nevertheless connected to a high number of them. The datasets containing the largest extended cliques were the MEN Test Collection [6] and the WordNet Synset Relatedness [5]. This resulted in sets of size 10 to 12, available at https://github.com/andrefs/grettir-datasets. Given the previously described difficulty on finding semantic datasets featuring named entities (which partially motivated this work), the cliques we obtained include concepts such as *floor*, *kitchen*, *staircase*, *ample* and *beam*.

We then proceeded to generate some test instances of Grettir featuring the items extracted from the cliques. These instances were used for the usability testing; in the end, we extracted the sorted list of pairs from each instance, with the goal of comparing them with the sorted lists of pairs corresponding to the cliques extracted from the semantic datasets.

6.2 Shooting Stars

The first instance of Grettir is named *Shooting Stars*, and it is focused on the relatedness of musical artists. It is publicly available at https://stars.andrefs.com.

In Shooting Stars we selected a list of 50 music artists and aimed to include widely known artists, from different musical genres, such as Bob Dylan, Rihanna, Édith Piaf or Elvis Presley. Players are asked *"Who is the intruder? Which of these 3 artists is the least related with the other two?"*. They are given no further instructions on which criteria should be used to compare the artists, and instructed to *"Click or tap on the intruder to select it!"*. They are also informed that if their opinion *"matches at least one of the other two players, you score! So, it's not really about your opinion but whether you can guess other people's opinions!"*. A screenshot of the main view of Shooting Stars can be found in Sect. 1, Fig. 1.

We publicized the games using a number of approaches, e.g. family and friends, social networks. The one which seemed to provide the best results was using a university mailing list, which by our estimates reached over 20k people, including students, faculty and other staff.

7 Results

This section describes the results obtained with the development of Grettir and the preliminary user testing and validation, and Shooting Stars, the first instance of Grettir.

7.1 Usability and Validation Tests

The answers to the usability questionnaire allowed us to identify and fix minor issues, most concerning small flaws or bugs in the user interface. The fact that these were computer science students means they were more knowledgeable regarding technical aspects of applications. There were no major usability issues found.

Regarding the validation tests, we did not managed to gather enough users to have data to order all the possible pairs in each instance. We were not able to extract any conclusions regarding our algorithms accuracy from these tests.

7.2 Shooting Stars

Up until the moment of writing this paper, Shooting Stars had 1001 different users, 653 of which played at least one game (a user profile is automatically created when a new user accesses the web page). A total of 1013 games were played and finished. This resulted in 12 different artists being paired with each other and a total of 13 pairs sorted. The list is updated and available for download at https://github.com/andrefs/grettir-datasets/stars-2021115.txt.

1. Elton John – Bruce Springsteen
2. Matt Bellamy (Muse) – Steven Tyler (Aerosmith)
3. John Lennon (The Beatles) – Prince
4. Steven Tyler (Aerosmith) – Prince
5. Steven Tyler (Aerosmith) – John Lennon (The Beatles)
6. John Lennon (The Beatles) – Elton John
7. Elton John – Madonna
8. Steven Tyler (Aerosmith) – Madonna
9. Adele – Eddie Vedder (Pearl Jam)
10. Madonna – Eddie Vedder (Pearl Jam)
11. Rihanna – John Lennon (The Beatles)
12. Rihanna – Matt Bellamy (Muse)
13. Matt Bellamy (Muse) – John Lennon (The Beatles)

Due to the algorithm used to build our dataset we cannot compare the sorted pairs established by each individual player. As such, we cannot calculate the inter-annotator agreement (IAG) – a common metric to evaluate the quality of human-generated datasets – by calculating the correlation between each players sorted pairs. Nevertheless, we calculated the IAG by averaging the consensus among all relevant triples (the number of players who agreed on the most voted entity divided by the total number of votes for that triple). We obtained an average IAG of 0.693 with a standard deviation of 0.183.

8 Discussion

Observing the user interaction and analyzing the answers to the usability questionnaire allowed to preemptively fix a few minor issues in the game interface.

The algorithm validation was inconclusive due to the low number of test subjects gathered.

With Shooting Stars, we were able to produce a list of pairs of musical artists sorted by their relatedness, but smaller that what we anticipated (13 pairs) due to the low number of total games played. The game was not addictive enough to attract more players and to have them playing more games. One reason for this low engagement might be the game domain. If this is the case, an instance about chess, football players or historical monuments might attract more players. Another reason might be the game mechanics. The lack of real-time human opponents, or the simple *find-the-intruder* approach might have hindered our expectations. Lastly, the game design, kept visually simple and lacking sound effects might also have played a part in these results.

Semantic relations datasets are usually built by requiring human participants to assign numeric values to relations between pairs of entities. Similarly to the previously mentioned MEN dataset [6], we wanted to build a dataset using another approach: asking participants to compare pairs of entities. This method requires more human interaction: even when minimizing pair comparisons by using an AVL tree, sorting n pairs requires asking users about $\mathcal{O}(n \log n)$ pairs, instead of the $\Theta(n)$ pairs the usual method would require. Nevertheless, our claim is that attributing a numeric value to a relationship can be difficult. This is specially true when that number must belong to a closed interval, and you do not know all the entities beforehand: how much related are Eddie Vedder and Rihanna? What if now you need to compare Rihanna and Mozart? We hoped that this problem of dataset construction by pair comparison might be more easily tackled by formulating it as a game; the results so far have been inconclusive.

The code written for Grettir and Shootings Stars is open source and published under a GNU GPLv3 license. You can find the code, links to all the different components, and installation instructions (including a Docker version) at https://github.com/andrefs/shooting-stars.

9 Conclusions

Semantic measures mimic the human capability of evaluating how similar or how related two things are. These measures are algorithms which extract semantic clues from a number of semantic sources. They are frequently used to compare pairs of concepts but frequently also instances. Semantic relations datasets can be used to evaluate such measures. These datasets are usually built by asking human annotators to explicitly attribute a numeric value to the strength of the relationship between two entities.

Semantic relations datasets can also be built by asking human annotators to compare pairs of entities against each other, and rank them. These comparisons are then used to sort the pairs of entities, producing a sorted list of pairs. This method requires more input from the annotators, but each question asked is more naturally answered.

Formulating this problem as a game made people contribute to the dataset construction which otherwise would probably not. Nevertheless, it did not provide the number of players or games played which would make this effort a success.

Grettir is a platform which can be used to implement other instances of this game, focused on other domains. The main effort in producing these other instances would be implementing a different interface, with a different design which would match the game lore.

Funding Information. André Santos has a Ph. D. Grant SFRH/BD/129225/2017 from Fundação para a Ciência e Tecnologia (FCT), Portugal.
This work is also financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.

References

1. Lin, D.: An information-theoretic definition of similarity. In: ICML 1998, pp. 296–304 (1998)
2. Metrology, J.: International vocabulary of metrology - basic and general concepts and associated terms (VIM). (BIPM Sèvres 2008) (2008)
3. Ziegler, C., Simon, K., Lausen, G.: Automatic computation of semantic proximity using taxonomic knowledge. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 465–474 (2006)
4. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web **9**, 77–129 (2018)
5. Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R.: Adding dense, weighted connections to WordNet. In: Proceedings of the Third International WordNet Conference, pp. 29–36 (2006)
6. Bruni, E., Tran, N., Baroni, M.: Multimodal distributional semantics. J. Artif. Intell. Res. **49**, 1–47 (2014)
7. Albertoni, R., De Martino, M.: Semantic similarity of ontology instances tailored on the application context. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 1020–1038. Springer, Heidelberg (2006). https://doi.org/10.1007/11914853_66
8. Pirró, G., Euzenat, J.: A feature and information theoretic framework for semantic similarity and relatedness. In: The Semantic Web-ISWC 2010, pp. 615–630 (2010)
9. Bodenreider, O., Aubry, M., Burgun, A.: Non-lexical approaches to identifying associative relations in the gene ontology. In: Pacific Symposium on Biocomputing, p. 91 (2005)
10. Ranwez, S., Ranwez, V., Villerd, J., Crampes, M.: Ontological distance measures for information visualisation on conceptual maps. In: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, pp. 1050–1061 (2006)
11. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S.: A word at a time: computing word relatedness using temporal semantic analysis. In: Proceedings of the 20th International Conference on World Wide Web, pp. 337–346 (2011)
12. Halawi, G., Dror, G., Gabrilovich, E., Koren, Y.: Large-scale learning of word relatedness with constraints. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1406–1414 (2012)

13. Barzegar, S., Davis, B., Zarrouk, M., Handschuh, S., Freitas, A.: SemR-11: a multilingual gold-standard for semantic similarity and relatedness for eleven languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018). https://aclanthology.org/L18-1618
14. Djaouti, D., Alvarez, J., Jessel, J.: Classifying serious games: the G/P/S model. In: Handbook of Research on Improving Learning and Motivation Through Educational Games: Multidisciplinary Approaches, pp. 118–136 (2011)
15. Jones, N., Brun, A., Boyer, A.: Comparisons instead of ratings: towards more stable preferences. In: 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, vol. 1, pp. 451–456 (2011)
16. Dörner, R., Göbel, S., Effelsberg, W., Wiemeyer, J.: Serious Games. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40612-1
17. Von Ahn, L.: Games with a purpose. Computer **39**, 92–94 (2006)
18. Von Ahn, L., Dabbish, L.: Designing games with a purpose. Commun. ACM **51**, 58–67 (2008)
19. Zar, J.: Spearman rank correlation. In: Encyclopedia of Biostatistics, vol. 7 (2005)
20. Freedman, D., Pisani, R., Purves, R.: Statistics (International Student Edition), 4th edn. WW Norton & Company, New York (2007)
21. Law, E., Von Ahn, L., Dannenberg, R., Crawford, M.: TagATune: a game for music and sound annotation. In: ISMIR, vol. 3, p. 2 (2007)
22. Dulačka, P., Šimko, J., Bieliková, M.: Validation of music metadata via game with a purpose. In: Proceedings of the 8th International Conference on Semantic Systems, pp. 177–180 (2012)
23. Fort, K., Guillaume, B., Chastant, H.: Creating zombilingo, a game with a purpose for dependency syntax annotation. In: Proceedings of the First International Workshop on Gamification for Information Retrieval, pp. 2–6 (2014)
24. Pearl, L., Steyvers, M.: Identifying emotions, intentions, and attitudes in text using a game with a purpose. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 71–79 (2010)
25. Celino, I., Re Calegari, G., Fiano, A.: Refining linked data with games with a purpose. Data Intell. **2**, 417–442 (2020)
26. Vannella, D., Jurgens, D., Scarfini, D., Toscani, D., Navigli, R.: Validating and extending semantic knowledge bases using video games with a purpose. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1294–1304 (2014)
27. Calegari, G., Fiano, A., Celino, I.: A framework to build games with a purpose for linked data refinement. In: International Semantic Web Conference, pp. 154–169 (2018)
28. Siorpaes, K., Hepp, M.: Games with a purpose for the semantic web. IEEE Intell. Syst. **23**, 50–60 (2008)
29. Siorpaes, K., Hepp, M.: OntoGame: weaving the semantic web by online games. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 751–766. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68234-9_54
30. Oguz, C., Blessing, A., Kuhn, J., Im Walde, S.: WordGuess: using associations for guessing, learning and exploring related words. In: Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021), pp. 235–241 (2021)
31. Kuittinen, J., Kultima, A., Niemelä, J., Paavilainen, J.: Casual games discussion. In: Proceedings of the 2007 Conference on Future Play, pp. 105–112 (2007)

32. Juul, J.: A Casual Revolution: Reinventing Video Games and Their Players. MIT Press, Cambridge (2010)
33. Chess, S., Paul, C.: The end of casual: long live casual. Games Cult. **14**, 107–118 (2019)
34. Mäyrä, F., Alha, K.: Mobile gaming. In: The Video Game Debate, vol. 2, pp. 107–120 (2020)
35. Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.: Semantic Similarity from Natural Language and Ontology Analysis. Synthesis Lectures on Human Language Technologies, vol. 8, pp. 1–254 (2015)
36. Nielsen, J.: Why you only need to test with 5 users. (Useit.com Alertbox) (2000)
37. Nielsen, J., Landauer, T.: A mathematical model of the finding of usability problems. In: Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems, pp. 206–213 (1993)