# A Random Walk Model for Entity Relatedness

Pablo Torres-Tramón and Conor Hayes

Insight Centre for Data Analytics
NUI Galway, Ireland
`pablo.torres@insight-centre.org`
`conor.hayes@insight-centre.org`

**Abstract.** Semantic relatedness is a critical measure for a wide variety of applications nowadays. Numerous models, including path-based, have been proposed for this task with great success in many applications during the last few years. Among these applications, many of them require computing semantic relatedness between hundreds of pairs of items as part of their regular input. This scenario demands a computationally efficient model to process hundreds of queries in short time spans. Unfortunately, Path-based models are computationally challenging, creating large bottlenecks when facing these circumstances. Current approaches for reducing this computation have focused on limiting the number of paths to consider between entities.
Contrariwise, we claim that a semantic relatedness model based on random walks is a better alternative for handling the computational cost. To this end, we developed a model based on the well-studied Katz score. Our model addresses the scalability issues of Path-based models by pre-computing relatedness for all pair of vertices in the knowledge graph beforehand and later providing them when needed in querying time. Our current findings demonstrate that our model has a competitive performance in comparison to Path-based models while being computationally efficient for high-demanding applications.

**Keywords:** Entity Relatedness · Path-based Semantics · Random Walks

## 1 Motivation

Graph-based semantic relatedness assessment between two entities has been applied to a wide verity of applications such as Word Sense Disambiguation (WSD) [3,16], Entity Search [9,22], Named Entity Disambiguation (NED) [20,16] and, more recently, the *cold start* problem found in recommender systems [21]. In all these cases, semantic relatedness serves two purposes: *i*) as a pre-processing step for regular input [3,16,20] or *ii*) as intermediary step in their processing pipeline [9,22,21]. For instance, entity search requires to identify a set of named entities that are semantically close to an original entity under certain user requirements. In this case, semantic relatedness is regarded as a proximity measure between pairs of entities that are under exploration. A similar situation occurs in NED as well. Here semantic relatedness is used to quantify the solution space such that the instance whose entities have the highest pairwise semantic relatedness is the solution to the disambiguation problem. The number of queries required by these applications in their normal activity can easily reach the thousands in short time spans. Therefore, these applications configure a scenario where the computational performance of semantic relatedness becomes critical.

Unfortunately, graph-based semantic relatedness models are computationally costly. The reason for this cost is rooted in the formalisation of these models. Most of them were designed as a ranking of aggregating path-based scores [22,12,16,20]. Thus, they require to enumerate a large number of paths between the input entities. Finding these paths is the cause of the high cost as this task is very expensive for even slightly dense graphs. Although knowledge graphs are far from being dense, the execution time of this model can be high in practice. Current attempts to improve the runtime performance of these models are focused on limiting the number of paths to enumerate [20,16,22] by setting an upper bound on the number of paths to find. However, this strategy fails to reduce the complexity as it does not set a bound on the plausible number of paths to check. Hence keeping the same complexity in the worst case.

Rather than enumerating paths, a more scalable alternative can be achieved by using random walks. A walk is a generalisation of paths such that cycles (or edge repetitions) are permitted. Such freedom is exploited extensively as certain linear algebra properties allow us to represent the walk finding problem as a linear equation system [17]. Surprisingly, walks are normally overlooked as a source of semantics. In this paper, we propose an algorithm for semantic relatedness using a random walk model based on the well-known Katz centrality [17]. We argue that random walks, in general, have been underestimated as a source of relatedness. Our work demonstrates that random walks can have a performance as good as direct paths while being substantially more efficient. We found that the relatedness scores generated by our model are ranking-equivalent to a well-known path-based model. Our model reduces dramatically the time required for processing a single query, being ideal for high-demanding applications.

The rest of this paper is structured in the following way: In the next section, we introduce our method, detailing its mathematical foundations as well as its properties. In the third section, a detailed description of the implementation is given. Later, in the fourth section, we evaluate our algorithm, comparing its performance against the state-of-the-art methods as well as their runtimes. Next, we give an account of the related work in section five. Finally, our conclusions and future work.

## 2 Method

Before presenting the proposed method, it is necessary to introduce a series of mathematical concepts required for the formalisation of our method.

### 2.1 Preliminaries

A knowledge base is a set of facts coded in the form of triples, which in turn are formed by a subject, an object, and a predicate. The intersection of subjects (or objects) between different triples allows us to chain triples around a common subject (or object). Naturally, this leads to viewing a knowledge base as a graph-like structure.

**Definition 1 (Knowledge Graph).** *Given a set of triples $T \in \mathbb{S} \times \Sigma \times \mathbb{O}$ such that $\mathbb{S} \cap \mathbb{O} \neq \emptyset$. A knowledge graph is a tuple $G = (\mathcal{V}, \mathcal{E}, \sigma)$ such that its vertices are defined by $\mathcal{V} = \mathbb{S} \cap \mathbb{O}$, its edges as $\mathcal{E} = \{(i,j) \;:\; (i,p,j) \in T\} \subseteq \mathcal{V}^2$, and $\sigma : \mathcal{E} \longrightarrow \Sigma$ is a map for an edge to the set of predicate labels $\Sigma$. For each triple $(i,p,j) \in T$ there is an equivalent inverse edge $\hat{e} = (j,i) \in \mathcal{E}$ such that its type is given by the type of the original edge ($\sigma(e) = \sigma(\hat{e})$). The resulting graph $G$ is strongly connected, i.e. any vertex $i$ is reachable from any other vertex $j$ and the adjacency matrix of the graph is symmetrical.*

The imposition of symmetry is a common practice in the field [16,22,7]. By setting it, we can traverse the graph in both directions having the same relationships defined in the original triple set. Semantically speaking, this implies that for every subject-object relationship, there is an equivalent-opposite relationship connecting the entities. The same situation occurs when considering a sequence of edges. If there is a sequence of edges connecting two entities, then there is an equivalent-opposite sequence of edges that connects the same pair of entities in the opposite direction. Therefore, it is easy to visualise that, for any two entities pairs, the number of sequences connecting them is the same. This assumption is crucial to the formalisation of the proposed method as it makes the adjacency matrix symmetrical.

**Definition 2 (Walk).** *Let $G = (\mathcal{V}, \mathcal{E}, \sigma)$ be a knowledge graph. A walk $W$ in $G$ is a finite, non-empty sequence of edges $e_1, e_2, \ldots, e_k \in \mathcal{E}$ connecting $v_1$ and $v_{k+1}$. The vertices $v_1 \in e_1$ and $v_{k+1} \in e_k$ are normally called as the initial and final vertices of the sequence respectively. The length of a walk (denoted as $|W|$) is indicated by the cardinality of the sequence.*

Basically, a walk starts at some given vertex and follows a certain sequence of edges until reaching the final vertex. In semantic terms, the length of a walk reflects the effort of moving along the graph from one entity to another. As the length increases, less pertinent become the sequence. In order to compare walk-based methods against paths-based ones, it is then necessary to formalise the latter as well.

**Definition 3 (Path).** *Let $G = (\mathcal{V}, \mathcal{E}, \sigma)$ be a knowledge graph. A path $P$ in $G$ is a walk connecting $v_1$ and $v_{k+1}$ such that there is no repetition of vertices in the sequence.*

Katz centrality [17] is a classical score from social network analysis that measures the overall influence for each vertex in the graph. The original formalisation was based on the geometric progression $(\beta A)^0 + (\beta A)^1 + (\beta A)^2 + \ldots$ for an adjacency matrix $A$ and real, positive value $\beta$ which converges under certain conditions. In this work, however, a formalisation based on random walks is used instead. A random walk model is a Markov chain such that the next vertex in the chain will be selected independently at any given time (or step) $k$, regardless of previously visited vertices. Katz centrality can also be formalised in this way.

**Definition 4 (Katz).** *Let $G$ be a knowledge graph and $A$ its adjacency matrix. Let $\beta \in (0,1] \subset \mathbb{R}$ be a given parameter. Let $D$ be a diagonal matrix such that $D_{(i,i)} = \sum_i (A_{:,i})$. The transition matrix $T$ is given by $T = D^{-1}A$. A Katz random walk process over $G$ is a Markov chain $M$ such that:*

$$M_{k+1} = \beta T M_k \tag{1}$$

*Where $k$ is the number of steps or time. The initial value of Katz $M_0 = I$, which is the identity matrix.*

Each pair $(i,j)$ of the resulting matrix $M_k$ is the probability of reaching vertex $j$ by randomly walking $k$ steps from vertex $i$. For each step taken, $\beta$ is a penalty value that reduces the influence of the walks in the final probability.

Knowing that $M_0 = I$, let us consider the following operator for this random walk process whenever a given value $t \in \mathbb{Z}_{>0}$ is given.

$$\Delta_0^t = \sum_{k=0}^{t} M_k = \sum_{k=0}^{t} (\beta T)^k \tag{2}$$

Here, only the probabilities of the walks formed for a certain time range are considered. Thus, for any $t$, $\Delta_0^t(i,j)$ is the probability of reaching vertex $j$ from vertex $i$ by randomly walking $t$ or less steps. It has been a well-extended practice in the field to only consider sequences up to a certain length, normally 4 [20,16] or 5 [22]. Therefore, this operator will be used to control the length of the walks. When $t \to \infty$, then $\Delta_0^t$ converges exactly to the Katz centrality. The convergence exists whenever $\beta$ is less than the reciprocal of the spectral radius $\rho$ of the transition matrix $T$, i.e. $\beta < 1/\rho(T)$.

## 2.2 Queries

Given a knowledge graph $G$ and certain input vertex $j$, the probability of randomly walking from any other vertex in the graph until reaching $j$ in $t$ or fewer steps can be written as it follows:

$$Pr(X_{\leq t} = j|X_0) = \sum_{k=1}^{t} Pr(X_k = j|X_0) Pr(S = 0)^k \tag{3}$$

Where $S$ is a uniform random variable and it represents the probability of a walk to finish at any vertex at random. The probability $Pr(X_k = j|X_0)$ is the combined probability of any random walks starting at any vertex at time 0 and reaching vertex $j$ at time $k$. Now, it is evident that if a walk reaches $j$, it must have a sequence of edges $(X_0, X_1), (X_1, X_2), \ldots, (X_{k-1}, X_k)$ such that $X_k = j$. Each unique vertex $X_l$ in this sequence is an independent random variable. Thus, the probability of a walk is given by the joint probability of its vertices in the sequence of edges. Therefore, the expression $Pr(X_0|X_k = j)$ can be written as follows:

$$Pr(X_0|X_k = j) = \left( \prod_{l=0}^{k-2} Pr(X_l|X_{l+1}) \right) Pr(X_{k-1}|X_k = j) \tag{4}$$

The expression $Pr(X_l|X_{l+1})$ (as well as $Pr(X_{k-1}|X_k = j)$) is the probability of randomly walk from $X_l$ to $X_{l+1}$ for any $l < k$. In the absence of data, we need to resort to heuristics in order to estimate $Pr(X_l|X_{l+1})$ for any pair $(l, l+1) \in \mathcal{E}$ assumming that this probabilities do not change for any pair $(l, l+1)$.

1. EQV: Each edge has the same probability.
2. PFITF [22]: The probability of a single edge is proportional to the amount of information contained by this edge at the local context and the frequency of the edge type across the entire graph. The local context for an edge is the set of edges incoming or outgoing to any of its two vertices.
3. EXCL [16]: Each edge probability is proportional to the level of the rareness of its type, considering only the local context.

These three heuristics defined the three different transition matrices that were used in the evaluation section. Once these probabilities are estimated, a ranking-preserving solution for $Pr(X_{<t} = j|X_0)$ can be computed directly using the Katz operator over the transition matrix.

$$Pr(X_{\leq t} = j|X_0) \propto \mathit{\Delta}_0^t(., j) \tag{5}$$

Where $(., j)$ means to select the column $j$ of the resulting matrix. This column contains the probability of randomly walking from any vertex to vertex $j$.

Now, if a sample set of pairs of entities is given, then a solution can be generated by the same principle. Suppose that $\tau$ is the set of target vertices and be $\iota$ the set of initial vertices defined by the sample set of pairs. The probability to estimate in this case is given by $Pr(X_{\leq t} \in \tau|X_0 \in \iota)$

$$Pr(X_{\leq t} \in \tau|X_0 \in \iota) \propto \mathit{\Delta}_0^t(i, j) \ \ \forall i \in \iota, \ j \in \tau \tag{6}$$

## 2.3 Properties

$\mathit{\Delta}_0^t$ is the basic operation for solving $Pr(X_{\leq t} \in \tau|X_0 \in \iota)$. Here a comparison between the ranking produced by this operator and a previous path-based relatedness called *Katz Relatedness* [20,16] is conducted since both are based on a similar principle. Before proceeding to compare their rankings, both scores will be re-defined in terms of a path and walk contribution respectively and presented as similarity (resemblance) measures.

**Definition 5 (Walk-based Relatedness).** *Let $G$ be a knowledge graph and $t \in \mathbb{Z}_{>0}$ the step parameter. Let $\hat{\Delta}_0^t$ be a normalisation operator $\hat{\Delta}$ such that $\hat{\Delta}_0^t$ is symmetric. The function $\phi : (i,j) \in \mathcal{E} \longrightarrow \mathbb{R}$ is called Walk Relatedness at $t$ such that:*

$$\phi(i,j) = \begin{cases} \hat{\Delta}_0^t(i,j) & \textit{if } i \neq j \\ 1 & \textit{otherwise} \end{cases} \tag{7}$$

**Proposition 1.** *Function $\phi$ is a similarity measure.*

*Proof.* It is necessary to prove that $\phi$ is symmetric and $\forall i,j \in \mathcal{E}, \quad \phi(i,i) \geq \phi(i,j)$. The first condition is trivial since $\hat{\Delta}_0^t$ is symmetric. For the second case, $\hat{\Delta}_0^t(i,j) \leq 1$ for every pair $(i,j) \;\; i \neq j$. Since $\phi(i,i) = 1$, then it is evident that the second condition holds.

As this definition required, it is necessary to introduce a normalisation for the resulting matrix of Katz operator (equation 2). To this end, the following normalisation is applied to the result of the Katz operator.

$$\hat{\Delta}_0^t = norm(\Delta_0^t + {\Delta_0^t}^T) \tag{8}$$

Where *norm* is the max-min normalisation. Notice that adding $\Delta_0^t$ with its transpose generates a symmetric matrix. This addition has also a semantic implication for our model. It reflects the probability of reaching $j$ from $i$ in both directions.

*Katz Relatedness* cannot be expressed as a geometric series of the transition matrix. Instead, it must be defined in terms of the paths connecting each pair $(i,j)$.

**Definition 6 (Path-based Relatedness).** *Let $G$ be a knowledge graph. Let $\mathcal{P}_{(i,j)} = \bigcup_{k=1}^{t} \mathcal{P}_{(i,j)}^k$ be the union of the set of paths connecting vertices $(i,j)$ in $G$ with length $k$. The function $\psi : \mathcal{V}^2 \longrightarrow [0,1] \subset \mathbb{R}$ is called Path-based Relatedness such that:*

$$\psi(i,j) = \begin{cases} \frac{1}{|\mathcal{P}(i,j)^t|} \sum_{k=1}^{t} \sum_{P \in \mathcal{P}_{(i,j)}^k} Pr(P)\beta^k & \textit{if } i \neq j \\ 1 & \textit{otherwise} \end{cases} \tag{9}$$

Notice that the operator $\Delta_0^t$ can also be written in terms of individual contributions of its constituents, similar to equation 3.

**Proposition 2.** *Function $\psi$ is a similarity measure.*

*Proof.* Each path connecting $(i,j)$ have a reciprocal path in the opposite direction, thus $\psi(i,j) = \psi(j,i)$. By definition, the relatedness between an entity and itself is $\psi(i,i) = 1$. It is evident then that $\psi(i,i) \geq \psi(i,j) \; \forall i,j \in \mathcal{E}$.

Naturally, both relatedness functions, $\phi$ and $\psi$, induce a partial order over the set of unordered pairs of vertices. Thus, the subsequent binary relationships are defined for any $i,j,k,l \in \mathcal{V}$:

$$\begin{aligned} (i,j) \preceq_\phi (k,l) &\iff \phi(i,j) \leq \phi(k,l) \\ (i,j) \preceq_\psi (k,l) &\iff \psi(i,j) \leq \psi(k,l) \end{aligned} \tag{10}$$

## 3 Implementation

So far, knowledge graphs have been formalised as a conventional graph with the particularity of having many relationship types. However, it is this very same particularity that makes knowledge graph a multi-graph. A multi-graph (or multi-dimensional graph) is a generalisation of a graph where each edge has a type associated. A single adjacency matrix cannot represent this type of graph. Instead, a collection of adjacency matrices, one for each relationship, is required. Thus, the adjacency matrix is indeed a tensor. A tensor is a generalisation of a matrix such that there is a third additional coordinate. For multi-graphs, this new coordinate represents the type of the relationship for a given pair of vertices $(i, j)$.

Evidently, tensors are more memory-consuming than matrices for storing data. Notice that, for each relationship, there is a $m \times m$ matrix to fill, requiring in total $m \times m \times k$ storage space. This situation makes it difficult to employ tensor models for implementing knowledge graphs in practice. One well-extended alternative to tensors is collapsing the data into a single adjacency matrix. Here, each element $(i, j)$ for each relationship $k$ is summed across the entire set of relationships. The following equation show the resulting adjacency matrix:

$$A(i, j) = \sum_{k=0}^{|k|} T(i, j, k) \tag{11}$$

Where $T(i, j, k)$ represents the adjacency for vertices $(i, j)$ and the type $k$. The resulting adjacency matrix only requires $m \times m$ storage space and it can be used safely to compute a geometric progression, producing the exact result. A more detailed proof of this technique can be consulted here [1].

Although this compression significantly reduces the overall memory consumption, it still requires $m \times m$ space in memory. For a large knowledge graph, this amount of space is still unfeasible. We observed, however, that the knowledge graph (and therefore the adjacency matrix) is quite sparse. Indeed, it was observed in the experiments that the sparsity level is about 99.9% for the knowledge graph employed. The level of the sparsity of a graph is intrinsically connected to its completeness. In general, knowledge graphs suffer a lack of completeness [13]. Thus, the number of relationships is often low. The implementation of our method takes advantage of this issue by storing the adjacency matrix as a sparse matrix. In sparse matrices, only non-zero values are stored, alleviating significantly the requirements of memory. Using sparse matrices makes the storage space needed by the adjacency matrix linear to the number of triples of the knowledge graph. We used the well-known Numpy [1] framework and its sparse package to implement our method.

Once the adjacency matrices are collapsed and stored as one matrix, the Katz operato (equation 2) can be computed directly from it. The high sparsity of the matrix is critical in this step to reduce the computation of the geometric

[1] `http://www.numpy.org`

**Algorithm 1:** Computing $\Delta_0^t$

```
  Data: $A$: collapsed adjacency matrix, $t$: number of steps, $\beta$
  Result: $\Delta_0^t$
1 m = size(A), D = eye(m), M = eye(m);
2 for $i < t$ do
3     M = M*($\beta$*A);
4     D = D + M;
5 end
```

progression. Since there is a high number of zeros in the matrix, the resulting matrices of the successive multiplications are also sparse. Therefore, the computation is dramatically reduced. As depicted in algorithm 1, the final number of multiplications is given by the length of the progression. In each iteration, the adjacency matrix is multiplied by itself and the parameter $\beta$ and stored for the next iteration. At the same time, the resulting multiplication is accumulated.

## 4 Evaluation

At the beginning of this paper, we argued that some real-world applications such as WSD [16] or recommender systems [21] require relatedness functions in order to rank a set of pairs. Hence, such ability will be tested using pairs of words as input in this evaluation. There are a few ground truth datasets that can be useful in this setting. Before going into their details, it is necessary first to describe the knowledge graph used in this evaluation. Later, we will compare human-produced rankings with the ones generated by our method and other state-of-the-art methods. Finally, a detailed account of the computational performance of the method will be provided.

### 4.1 WordNet

Since we are basing our evaluation in rankings of pairs of words, it is then necessary to select a Knowledge Graph according to these requirements. Among the available alternatives, we found that WordNet [2] presented two important features for our evaluation: *i*) it is relatively small in comparison to general purposes knowledge databases and *ii*) it is a good fit when analysing of pairs of words. Due to the cost of multiplying large matrices, we opted for selecting a relatively small knowledge graph. We left for the future the exploration of alternatives methods to reduce the amount of computation.

WordNet was introduced as a general purpose lexical database in the beginnings of the 90s [18]. In WordNet, words (referred as lemmas) are linked to abstract entities, known as concepts or *synsets*, which in turn are inter-linked among themselves with certain semantic relationships. Each relationship has a

[2] `https://wordnet.princeton.edu`

type associated that represent the semantic associations between both concepts. A concept might have several lemmas linked to it and vice-versa. Concepts are abstract ideas/objects that are evoked by a word when is used in a certain context or predicate. For example, lemmas: *car* and *automobile* normally refer to self-propelling objects that generally use a combustion engine to self-propel. However, other uses of the word *car* are valid as well in different contexts. For instance, a *car* might refer to a train wagon. As we mentioned, concepts are inter-linked according to some semantic relationships. For example, one concept might be part of a more general concepts (e.g. *car* $\longrightarrow$ *vehicle*) or they can refer to opposite ideas (e.g. *good* $\longrightarrow$ *bad*). WordNet represents these semantic associations using a fixed set of relationship types. The latest version of WordNet, (WN31), contains 26 relationship types.

The resulting graph generated from these concepts and its relationships was used for computing the Katz operator. During this process, it was found that some few concepts do not have any edge to other concepts in the graph. Therefore, these were removed from the graph as we assumed that this must be strongly connected. In summary, the resulting graph was composed by 116,787 vertices and 378,203 relationships, with an average degree of 6.476.

## 4.2 Ground Truth Datasets

Ground Truth datasets are composed by a list of pairs of words, where each pair have a real, positive score associated. These scores were granted by a set of human assessors, who evaluateed the degree of semantic association between the words. The scores determine the position of the pair in the ranking. It is assumed that in the input list, the pairs of words do not necessarily must have words in common. Instead, we assumed that each pair was generated independently of the others, discarding any relationship among themselves.

In the literature, there are several ground truth datasets that fit this purpose. Among these, the following three were selected:

1. MC [19] (28 pairs of words): This dataset was designed to investigate the semantic and the contextual similarity for words.
2. RG [23] (65 pairs of words): In this classical dataset, the authors explored the strength of synonymy between pairs of words. The list of pairs was composed by 65 pairs, including highly related and unrelated pairs.
3. WS-sim [2] (97 pairs of words): This dataset is a subset of the original dataset proposed here [11]. Agirre *et al.* claimed that the original dataset contained pairs of words for similarity and relatedness, and thus they divided the pairs in two groups: similarity and relatedness pairs. In this evaluation, only similarity pairs were used.

The words in these datasets are presented in plain text format. There is no information about the context in which they were present originally to the human assessors, or any other information that can help us to determine the sense of the word. Therefore, it is necessary to disambiguate these words to determine their senses. Fortunately, Schwartz *et al.* [24] already completed this step. In their work, they labelled the input words using WordNet concepts, obtaining a *synset* label for each word. It was necessary to drop some pairs of words from the datasets to produce a consistent label assignment as there were some cases where it was not possible to determine the WordNet concept suggested by the words.

## 4.3 Evaluation Metrics

In order to evaluate our rankings with respect to the one produced by human assessors, we used Spearman correlation. This metric measures the correlation between two random variables, generating an output value that ranges in the interval $[-1, 1]$. When there is a positive correlation then the output value would be close to 1. Instead, if there is a negative correlation then the output would be near to $-1$. When there is no correlation, the output is close to 0.

Before comparing the outputs, the scores produced by our algorithm as well as the scores assessed by human assessors were transformed into positions. If two pairs or more had an equal score, the same position number was assigned to all of them. In order to decide whether two scores were equal, we defined a value $\epsilon > 0$ such that if $|\phi(i,j) - \phi(l,k)| < \epsilon$ then the scores in question were considered to be equal. Since equal pairs have the same position in the ranking, the next pair started at the position indicated by the last position assigned plus the number of pairs holding that position. This transformation was done for any ranking evaluated. The evaluation metrics were then computed using these lists of positions as the input.

## 4.4 Discussion

We proceeded to compute the Katz operator for the following values: $t \in [2, 3, 4]$ using WN31 as the knowledge graph and applying the 3 different weighting schemes. The Spearman correlation between the rankings generated for the ground truth datasets and the human-produced rankings are shown in Table 1.

We observed that none of the weighting schemes is superior to any other for every dataset. Instead, each dataset has a scheme whose performance is better than the rest. For instance, excl weighting scheme has the best performance for WS-sim dataset. However, it performs relatively poor in the others. eqv and ptitf have a similar performance for RG, but eqv is significantly better than ptift when ranking MC.

We noted as well that the best results were achieved whenever $\beta \in [0.5, 1]$ for every datasets MC and RG. Only dataset WS-sim showed a more balanced distribution when using exclusivity scheme, being $\beta = 0.5$ the setting with the best performance. This is consistent with *de facto* value in random processes for this parameter ($\beta = 0.85$) [2]. As $\beta \longrightarrow 1$, the contributions of larger paths become more relevant in the final score. Therefore, we can deduce that the inclusion of these paths significantly increases the performance of the relatedness. For instance, the best result for MC was obtained when $\beta = 1.0$.

In the case of the number of steps ($t$), we observed that the best performances are well distributed across the range examined in this evaluation. We also considered the case of $t = 1$ (not displayed for the sake of space), finding the performance substantially poorer than any other case. Evidently, the inclusion of larger paths can dramatically increase the performance, particularly $t = 2$ for datasets MC and RG. However, after reaching a peak, the inclusion of larger paths did not increase necessarily the correlation. It is interesting to note that in every scheme examined, the performance was slightly similar when $t \in [2, 3]$ for MC and RG, being $t = 2$ marginally better. In contrast, the difference between the results obtained for WS-SIM using the same range of $t$ were significantly larger, obtaining better performance when $t = 3$.

The correlations for $\phi$ and $\psi$ were very similar for the analysed datasets as we expected. It was necessary to set $\beta$ in the range of $[0.05, 0.2]$ as larger values performed poorly in this case. This situation was also observed before [16].

### 4.5 Runtime

We evaluated the execution runtime for: *i*) generating $\Delta_0^t$ (Figure 1) and *ii*) solving queries using $\phi$ and $\psi$ (Figure 2). The measurements were conducted in a 24 core machine with 100 GB of RAM on Ubuntu 12.10. The query runtimes were obtained by measuring the wall time needed to solve each query in our datasets using different configurations. In total we executed $3,800$ instances for each function.

As we expected, the cost of computing $\Delta_0^t$ increased quickly as the number of steps did. However, the average of runtime for the maximum $t$ tested was less than 100s. Thus, commodity hardware is more than enough for computing this operator for the range examined. The cause of these low runtimes is due to the high sparsity of the transition matrix. This dramatically reduces the number of operations needed when computing the geometric progression.

For the case of query evaluation, the results show that the average time required for solving a single query using $\psi$ increased exponentially as the steps did. The runtimes show that some of these queries were very costly to handle using this model. For instance, there were cases that required more than $10^t$ ms to complete a single query. The function $\psi$ was implemented using a simple BFS algorithm with some minor optimisations. Although optimising BFS is out of the scope of this paper, it is worth to mention that there are many optimisations for BFS that can be used to improve the performance of $\psi$ [10].

In contrast, the computational cost was close to constant for single queries using $\phi$. A relevant proportion of the queries required less than 1ms in order to get completed.

**Table 1.** Spearman Correlation of our method with different parameters ($\beta$, t, weighting scheme) for each dataset

| Method | Weight | $\beta$ | MC | | | RG | | | WS-SIM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $t = 2$ | $t = 3$ | $t = 4$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 2$ | $t = 3$ | $t = 4$ |
| $\phi(i, j)$ | EQV | 0.25 | 0.788 | 0.788 | 0.788 | 0.780 | 0.785 | 0.785 | 0.568 | 0.626 | 0.625 |
| | | 0.50 | 0.788 | 0.788 | 0.763 | 0.780 | 0.785 | 0.779 | 0.570 | 0.632 | 0.613 |
| | | 0.75 | 0.788 | 0.788 | 0.774 | 0.780 | 0.785 | 0.782 | 0.571 | 0.635 | 0.621 |
| | | 1.00 | 0.788 | 0.790 | 0.801 | 0.780 | 0.786 | 0.794 | 0.570 | 0.640 | 0.620 |
| | PFITF | 0.25 | 0.796 | 0.795 | 0.795 | 0.782 | 0.785 | 0.788 | 0.552 | 0.598 | 0.591 |
| | | 0.50 | 0.796 | 0.795 | 0.795 | 0.782 | 0.785 | 0.792 | 0.553 | 0.614 | 0.596 |
| | | 0.75 | 0.796 | 0.796 | 0.772 | 0.782 | 0.785 | 0.780 | 0.551 | 0.614 | 0.589 |
| | | 1.00 | 0.796 | 0.796 | 0.767 | 0.782 | 0.785 | 0.775 | 0.550 | 0.615 | 0.590 |
| | EXCL | 0.25 | 0.785 | 0.764 | 0.764 | 0.780 | 0.786 | 0.786 | 0.578 | 0.627 | 0.629 |
| | | 0.50 | 0.785 | 0.785 | 0.785 | 0.780 | 0.784 | 0.787 | 0.580 | 0.645 | 0.630 |
| | | 0.75 | 0.785 | 0.785 | 0.785 | 0.780 | 0.784 | 0.790 | 0.578 | 0.644 | 0.621 |
| | | 1.00 | 0.785 | 0.779 | 0.755 | 0.780 | 0.782 | 0.778 | 0.577 | 0.644 | 0.616 |
| $\psi(i, j)$ | EQV | 0.05 | 0.782 | 0.795 | 0.784 | 0.781 | 0.781 | 0.786 | 0.552 | 0.598 | 0.577 |
| | | 0.10 | 0.782 | 0.795 | 0.785 | 0.781 | 0.781 | 0.787 | 0.542 | 0.587 | 0.552 |
| | | 0.15 | 0.782 | 0.795 | 0.765 | 0.781 | 0.781 | 0.779 | 0.528 | 0.575 | 0.521 |
| | | 0.20 | 0.782 | 0.795 | 0.710 | 0.781 | 0.781 | 0.747 | 0.520 | 0.538 | 0.407 |
| | PFITF | 0.05 | 0.787 | 0.787 | 0.769 | 0.779 | 0.779 | 0.785 | 0.524 | 0.575 | 0.536 |
| | | 0.10 | 0.787 | 0.794 | 0.782 | 0.779 | 0.779 | 0.785 | 0.508 | 0.555 | 0.498 |
| | | 0.15 | 0.787 | 0.799 | 0.781 | 0.779 | 0.776 | 0.777 | 0.496 | 0.539 | 0.451 |
| | | 0.20 | 0.787 | 0.788 | 0.735 | 0.779 | 0.763 | 0.750 | 0.478 | 0.497 | 0.376 |
| | EXCL | 0.05 | 0.782 | 0.778 | 0.787 | 0.780 | 0.781 | 0.789 | 0.576 | 0.625 | 0.621 |
| | | 0.10 | 0.782 | 0.752 | 0.765 | 0.780 | 0.773 | 0.784 | 0.575 | 0.622 | 0.613 |
| | | 0.15 | 0.782 | 0.753 | 0.769 | 0.780 | 0.767 | 0.773 | 0.570 | 0.614 | 0.602 |
| | | 0.20 | 0.782 | 0.747 | 0.753 | 0.780 | 0.759 | 0.758 | 0.569 | 0.606 | 0.576 |

## 5 Related Work

Entity relatedness is a well-studied problem [5,4,16,20,22,9,8,6]. The origins of the field are rooted in computational linguistics [6], where for many years functions for assessing semantic relatedness have been developed. The emergence of large knowledge graph during the 90s and earlier 2000s triggered the development of graph-based models [5,22,9,16,20]. Here, the semantics between entities is determined according to sequences connecting the entities. Other forms of semantics under this approach included isomorphism of sequences and join operations between them.

Depending on the application, graph-based models can be classified into two broad categories. On the one hand, semantic relatedness is used as a tool for conducting an exploratory search over a knowledge graph. In this scenario, a single user wants to find hidden connections that are neither obvious nor intuitive from the relationships themselves. Thus, the main objective of semantic relatedness

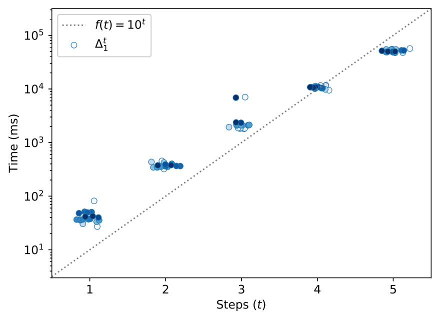

**Fig. 1.** Execution runtime for $\Delta_0^t$ using a range of $t$.

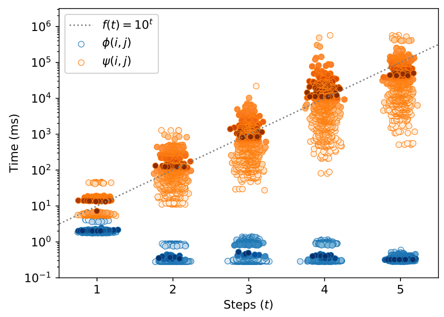

**Fig. 2.** Execution runtime for $\phi$ and $\psi$ using a range of $t$

here is to generate a minimum, connected subgraph that explains why two (or more) entities are related [22,9]. Different techniques have been employed to this end, including path covering in graphs [9] and SPARQL query listing [22] Optionally, paths composing the resulting subgraph can be ranked individually when the subgraph is large. These rankings of paths aim to display to the user the most relevant paths that integrate the subgraph [22]. An important number of applications have been developed for this use-case [15,8].

On the other hand, semantic relatedness is regarded as a ranking function. Here, the need is centred on the partial order generated by the semantic relatedness function [4,16,20,21]. The resulting ranking is then used as the input for another problem (e.g. WSD [16] or recommender systems [21]). Nunes *et al.* [20] proposed a ranking function that employed a mixture of textual and graph-based scores. Their graph score quantified the walks connecting a pair of entities in a similar fashion as Katz measure does. However, they restricted the type of walks considered in their score, allowing only paths. This idea was taken by Hulpuş *et al.* [16] where the paths were further restricted, using the top-$k$ shortest paths only. In both cases, the restriction of paths marginally alleviates the computational requirements. Our method instead, pre-computes the relatedness for any pair of vertices using random walks, solving queries in constant time.

Semantic relatedness models based on random walks have also been applied before. Agirre *et al.* [3] introduced a method based on the cosine similarity between two vectors that represent the Personalised Page Rank (PPR)for each word. Their method required to define a context, that later is used for determining the initial weight of PPR. In contrast, our method does not require any context or any other additional data. Moreover, it only requires computing the progression a single time, thus being more efficient. Gentile *et al.* [14] used a random walk model inspired in eigencentrality to derive a semantic relatedness score for concepts. They employed a parameter $t$ to control the length of the walks in a similar fashion as our approach. However, they set this parameter to a fixed value ($t = 2$) as opposite to our method in which it is variable.

## 6 Conclusions

In this work, a method for entity relatedness based on the Katz centrality has been introduced. The proposed method is significantly more efficient than state-of-the-art methods based on graph properties for the same use-case scenario. While being more efficient, the proposed method has a similar performance than state-of-the-art methods, being an effective solution when a large number of entity rankings is demanded. As a drawback, our method requires theoretically $O(m^2)$ space in order to store the relatedness scores. However, the high sparsity of $\varDelta_0^t$ suggests that this setback is much smaller in practice.

If the thechnique is to prove useful, a more efficient storage solution needs to be designed. Based on the expriments outlined here, approximated distances matrix appeared to be a reliable option for this end. Our evaluation only incorportated a relatively small knowledge graph (WordNet), therefore it is espically relevant to evaluate the peformance on larger knowledge graphs, such as BabelNet [3] or DBpedia [4]. Additionally, we hope to conduct a more robust evaluation to validate the rankings produced by this method and those generated by human assessors. Finally, alternative random walk models present a good opportunity for exploring alternatives to Katz centrality, for instance pagerank.

**Acknowledgements:** This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant No. SFI/12/RC/2289, co-funded by the European Regional Development Fund

## References

1. Acar, E., Dunlavy, D.M., Kolda, T.G.: Link prediction on evolving data using matrix and tensor factorizations. In: Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on. pp. 262–269. IEEE (2009)
2. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and wordnet-based approaches. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp. 19–27. Association for Computational Linguistics (2009)
3. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. pp. 33–41. Association for Computational Linguistics (2009)
4. Aleman-Meza, B., Halaschek, C., Arpinar, I.B., Sheth, A.P.: Context-aware semantic association ranking (2003)
5. Anyanwu, K., Sheth, A.: $\rho$-queries: enabling querying for semantic associations on the semantic web. In: Proceedings of the 12th international conference on World Wide Web. pp. 690–699. ACM (2003)

[3] https://babelnet.org
[4] https://wiki.dbpedia.org

6. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics **32**(1), 13–47 (2006)
7. Cheng, G., Shao, F., Qu, Y.: An empirical evaluation of techniques for ranking semantic associations. IEEE Transactions on Knowledge and Data Engineering **29**(11), 2388–2401 (2017)
8. Cheng, G., Zhang, Y., Qu, Y.: Explass: exploring associations between entities via top-k ontological patterns and facets. In: International Semantic Web Conference. pp. 422–437. Springer (2014)
9. Fang, L., Sarma, A.D., Yu, C., Bohannon, P.: Rex: explaining relationships between entity pairs. Proceedings of the VLDB Endowment **5**(3), 241–252 (2011)
10. Filtz, E., Savenkov, V., Umbrich, J.: On finding the k shortest paths in rdf data. In: Proceedings of the 5th International Workshop on Intelligent Exploration of Semantic Data (IESD 2016) co-located with the 15th International Semantic Web Conference (ISWC 2016). vol. 18 (2016)
11. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., uppin, E.: Placing search in context: The concept revisited. In: Proceedings of the 10th international conference on World Wide Web. pp. 406–414. ACM (2001)
12. Fionda, V., Pirrò, G.: Meta structures in knowledge graphs. In: International Semantic Web Conference. pp. 296–312. Springer (2017)
13. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 375–383. ACM (2017)
14. Gentile, A.L., Zhang, Z., Xia, L., Iria, J.: Semantic relatedness approach for named entity disambiguation. In: Italian Research Conference on Digital Libraries. pp. 137–148. Springer (2010)
15. Heim, P., Lohmann, S., Stegemann, T.: Interactive relationship discovery via the semantic web. In: Extended Semantic Web Conference. pp. 303–317. Springer (2010)
16. Hulpuş, I., Prangnawarat, N., Hayes, C.: Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In: International Semantic Web Conference. pp. 442–457. Springer (2015)
17. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
18. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to wordnet: An on-line lexical database. International journal of lexicography **3**(4), 235–244 (1990)
19. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and cognitive processes **6**(1), 1–28 (1991)
20. Nunes, B.P., Dietze, S., Casanova, M.A., Kawase, R., Fetahu, B., Nejdl, W.: Combining a co-occurrence-based and a semantic measure for entity linking. In: Extended Semantic Web Conference. pp. 548–562. Springer (2013)
21. Piao, G., Breslin, J.G.: Measuring semantic distance for linked open data-enabled recommender systems. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. pp. 315–320. ACM (2016)
22. Pirrò, G.: Explaining and suggesting relatedness in knowledge graphs. In: International Semantic Web Conference. pp. 622–639. Springer (2015)
23. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Communications of the ACM **8**(10), 627–633 (1965)
24. Schwartz, H.A., Gomez, F.: Evaluating semantic metrics on tasks of concept similarity. In: Cross-Disciplinary Advances in Applied Natural Language Processing: Issues and Approaches, pp. 324–340. IGI Global (2012)