Fast, Accurate, Multilingual Semantic Relatedness Measurement Using Wikipedia Links

Dante Degl'Innocenti, Dario De Nart, M. Helmy and C. Tasso

Abstract In this chapter we present a fast, accurate, and elegant metric to assess semantic relatedness among entities included in an hypertextual corpus building an novel language independent Vector Space Model. Such a technique is based upon the Jaccard similarity coefficient, approximated with the MinHash technique to generate a constant-size vector fingerprint for each entity in the considered corpus. This strategy allows evaluation of pairwise semantic relatedness in constant time, no matter how many entities are included in the data and how dense the internal link structure is. Being semantic relatedness a subtle and somewhat subjective matter, we evaluated our approach by running user tests on a crowdsourcing platform. To achieve a better evaluation we considered two collaboratively built corpora: the English Wikipedia and the Italian Wikipedia, which differ significantly in size, topology, and user base. The evaluation suggests that the proposed technique is able to generate satisfactory results, outperforming commercial baseline systems regardless of the employed data and the cultural differences of the considered test users.

Keywords Semantic networks · Vector space · Text processing theory Multilinguality

Mathematics Subject Classification (2010) Primary 68T30 · Secondary 68T50

D. Degl'Innocenti (✉) · D. De Nart · M. Helmy · C. Tasso
Department of Mathematics and Computer Science, University of Udine,
Via delle Scienze 206, Udine, Italy
e-mail: deglinnocenti.dante@spes.uniud.it

D. De Nart
e-mail: dario.denart@uniud.it

M. Helmy
e-mail: alameldien.muhammad@spes.uniud.it

C. Tasso
e-mail: carlo.tasso@uniud.it

K. Shaalan et al. (eds.), *Intelligent Natural Language Processing: Trends and Applications*, Studies in Computational Intelligence 740,
https://doi.org/10.1007/978-3-319-67056-0_27

1 Introduction

Measuring semantic likeness between items such as words, texts, or DBpedia entities is a vital component of several Artificial Intelligence applications, supporting tasks such as question answering, ontology alignment, Word Sense Disambiguation, and exploratory search. The concept of semantic likeness over the years has attracted the interest of the Natural Language Processing (NLP), Semantic Web, and Information Retrieval (IR) communities [8]. Two variants have been thoroughly discussed: *Semantic Similarity* which can be defined as the likeness of the meaning of two items, for instance "king" and "president" though not being synonyms have an high semantic similarity because they share the same function, and *Semantic Relatedness* which can be considered as a looser version of semantic similarity since it takes into account any kind of relationship, for instance "king" is semantically related to "Nation" because a king rules over a nation. Due to the high ambiguity of the very definition of these semantic relationships it is not uncommon to evaluate similarity and relatedness metrics upon their performance in a specific, well-defined and reproducible task [3].

Many metrics have been introduced in the literature, surveyed in Sect. 2, relying mostly upon word distribution or graph traversing over linked data. Such approaches, however present several shortcomings, most notably their evaluation tends to be demanding from a computational point of view, thus preventing their usage in scenarios where a very large number of comparisons must be made.

In this work we tackle the problem of assessing the degree of semantic likeness between entities from an exploratory search point of view, i.e. with the goal of retrieving for a given entity a neighbourhood of other entities which might be relevant from the point of view of an user who wants to learn more about the searched entity. With this task in mind, we focus on assessing semantic relatedness rather than semantic similarity.

We introduce therefore a new strategy to assess semantic relatedness between entities leveraging the link structure of a corpus of hypertextual documents. The employed similarity metric relies on the Jaccard similarity coefficient and exploits the Minhash optimisation to perform dimensionality reduction and therefore allow an efficient relatedness assessment. The presented model is then trained on both English and Italian Wikipedia and benchmarked against Google's and Bing's exploratory search tools which rely primarily on DBpedia and Freebase, allowing the comparison of search results. Our contribution is twofold: we introduce an efficient strategy to evaluate semantic similarity and we assess its performance upon the task of related entity retrieval over two distinct data sets written in different languages.

2 Related Work

A broad range of measures for assessing similarity and relatedness between entities has been proposed in the literature; such measures are grounded into set theory [19], statistics [22], and graph theory [18]. One of the best known semantic relatedness measures is the *Google Distance* [5] which exploits a search engine to estimate pairwise similarity between words or phrases. Such a metric has proven to be effective for a number of knowledge intensive tasks such as evaluating approximate ontology matching [7]. However the implied intensive usage of the underlying search engine makes this metric impractical or too expensive for most applications. Other strategies rely on structured knowledge bases such as taxonomies and ontologies. Wordnet[1] is among the first and still most used resources to estimate semantic similarity with a variety of techniques including graph search algorithms and machine learning. An extensive survey of semantic similarity metrics built upon Wordnet is presented in [3, 4]. The LOD cloud has also been widely exploited and several authors proposed strategies to evaluate similarity and relatedness among entities included in such a cloud. Most LOD-based techniques rely on the selection of a limited number of features among the multitude of properties present in the cloud, to perform this task techniques such as Personalized Page Rank are commonly used in the literature [18]. These techniques, despite being particularly demanding from a computational point of view are often used in the field of semantic-based personalisation [16]. Wikipedia has been used as well to compute semantic relatedness metrics: the authors of [6] introduce Explicit Semantic Analysis (ESA), a technique using machine learning to build vectorial representations of Wikipedia items based upon the textual contents of their corresponding articles. The authors of [23] propose an alternative to ESA which leverages the links included in Wikipedia articles to achieve similar performance but at a sensibly lower cost both in terms of computational complexity and of required data. The similarity metric therein presented is the combination of two metrics, one for incoming links and one for outgoing ones, the former one being closely related to the aforementioned Google Distance.

Vector Space Model (herein VSM) approaches are an alternative to explicit and formal knowledge representations such as the one provided by LOD. In a VSM entities, instead of being described by a set of predicates, are represented as a vector in a space with a finite number of dimensions. VSM leverage the *distributional hypothesis* of linguistics, which claims that words that occur in similar contexts tend to have similar meanings [9]. Some authors [17] in fact define the meaning of a concept as the set of all propositions including that concept. VSMs are commonly used to support several NLP and IR tasks, such as document retrieval, document clustering, document classification, word similarity, word clustering, word sense disambiguation, and many others. The most notable advantage of these techniques over formal representations is that vector spaces can be built in a totally automated and unsupervised way. For a deeper and more exhaustive survey of vector spaces and their usage in state of the art systems, we address the interested reader to [20], [15], and [14].

[1] http://wordnet.princeton.edu.

3 Similarity Assessment and Neighborhood Retrieval

As shown by the authors of [23] hypertextual connections between Web pages alone can carry a great deal of semantics at a reasonable computational cost. However their proposed method involving the combination of two distinct metrics for incoming and outgoing links can be still too demanding when a very large content base must be scouted to find related items. Wikipedia, which includes over 8 million items is a perfect example of such a situation. To overcome this limitations and to set up a minimal theoretical framework, we introduce a new hypothesis: the *Reference Hypothesis*. We assume that entities that are referenced in a similar set of documents might yield strong semantic affinity. For instance, in Fig. 1 two entities (*A* and *B*) are referenced by three different documents: this implies a semantic affinity between A and B.

This assumption is motivated by the fact that intuitively referencing something in a document implies the referenced item to be relevant in the context of the document, therefore entities that get constantly referenced together are relevant in the same contexts, hence they might be semantically related. This hypothesis can be seen as a generalised version of the aforementioned distributional hypothesis, however we would like to stress how even though words can be seen as entities, entities can be intended as way more abstract items, for instance other documents or ontology entries. For instance, the reference hypothesis applies to the scientific literature since articles citing similar sources are very likely to deal with similar topics. Other works in literature embrace this assumption though not formalising it, such as [10] wherein a scientific paper recommender system exploiting co-citation networks is presented. Building a vector space exploiting the Reference Hypothesis is straightforward once a large enough corpus of documents annotated with hyperlinks is provided. Within the corpus, two sets must be identified: the *entity set E* and the *document set D*; the first includes all the referenced entities, while the latter is the considered annotated documents. The vector space is represented with an $E \times D$ matrix that initially is a zero matrix. Iteratively, for each $d \in D$ all the references to elements in E are considered, and for each $e \in E$ referenced in d, the (e, d) cell of the matrix is set to 1. Since referencing a given entity only once in a document is a typical best practice

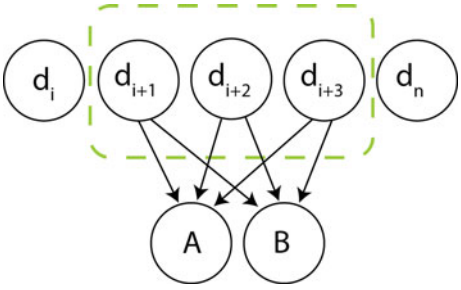


Fig. 1 Two entities referenced by the same set of documents

in several domains[2] we are not considering how many times e is referenced in d. Once all documents are processed we obtain a matrix where each row represents all the references to a given entity: we call such matrix *Reference Matrix* and the vector space it generates *referential space*.

Evaluating the similarity of two entities in such a vector space reduces to computing the distance between their vectors. Countless distance metrics exist in the literature such as norms, cosine similarity, hamming distance, and many others surveyed in [21]. All these metrics can be used in the Reference Matrix, however we prefer the Jaccard similarity coefficient (also known as Jaccard index [11]), defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{3.1}$$

where A and B are sets of items. Since each entity $e_i \in E$ can be considered a binary vector, it can also be expressed as the set that contains all the document $d_j \in D$ such that $(e_i, d_j) = 1$ in the Reference Matrix. The similarity of two equal sets is one, whereas the similarity between two sets that have no elements in common is zero. This choice is motivated by the intimate simplicity of such a metric and by the evidence presented in the literature that the Jaccard index performs better than other methods for finding word similarities in VSM approaches [12, 15].

However, evaluating the Jaccard index is linear in the size of the considered vectors, which can be extremely large when considering large corporas such as Wikipedia. The computation of the Jaccard index can be reduced to constant time using the MinHash optimisation [2]. Such a technique allows to efficiently compute the similarity between sets without explicitly computing their intersection and union. Its most common form consists in using an hash function to map each element of the set to an integer number and then selecting the minimum as a representative of the whole set. The probability that two different sets share the same minimum with respect to the hash function tends to the Jaccard similarity coefficient between the two sets [13]. The more hash functions are used, the closer the estimate gets to the real Jaccard similarity coefficient value. In this work we used 256 distinct hash functions to achieve a fine enough approximation of the Jaccard similarity coefficient. This translates to representing each entity as a 256 positions vector. Such a vector can be considered as an entity's fingerprint in the considered text corpus and implies a significant dimensionality reduction with respect to the initial vectorial space which may count millions of dimensions. This optimisations allows our method to scale up as the number of considered entities grows: being the number of positions of the fingerprint vector constant, checking semantic similarity between two entities will take constant time. With respect to other solutions presented in the literature such as [23] wherein the evaluation of semantic similarity is polynomial with respect to the size of the considered knowledge base, the MinHash optimisation significantly reduces the complexity of such an operation. As a matter of fact, checking which items are

[2]For instance in Wikipedia only the first time an entity is referenced it is annotated with an hyperlink, and in literature bibliographies have no duplicate entries.

the closest ones to a given entity implies checking the target entity against all items present in the knowledge base. With our solution this operation is linear with respect to the knowledge base's size, with other solutions it is quadratic in the best case.

4 Task Based Evaluation

Similarly to [3], we evaluated our system upon a specific application, in this case the retrieval of a set of neighbour entities for exploratory search purposes. Our evaluation activity, due to the intrinsic subjectivity of the very concept of semantic relatedness, was user-based. Two experiments are presented: in the first one we asked users to give an overall ranking to a list of related items, while in the second one we asked users to assess the relatedness of each item included in a given list to a target entity. Such an evaluation was performed over two datasets with different characteristic features and with two substantially different user groups to test the effectiveness of our methodology in different situations, thus preventing data overfitting and cultural biases in the presented conclusions.

4.1 Experimental Setting

Two hyperlinked text corpora were considered: the English Wikipedia and the Italian Wikipedia. The English Wikipedia is a well known and massive collaborative encyclopedia, counting over 8 million articles contributed by users from all around the world. On the other hand, the Italian one is a substantially smaller corpus, counting around 1.3 million articles and curated by users that mostly reside in Italy. We considered these two dataset because they differ significantly in size, in language, and in the user base that generated them.

Using the technique described in Sect. 3, a testbed system, herein named Referential Space Model (RSM), was developed and trained on Wikipedia, associating to each of its items a representative vector. Building on the results of [23] that provides evidence of the importance of both incoming and outgoing links, we also developed an alternative model relaxing the distributional hypothesis and considering outgoing links, i.e. the items mentioned in the article corresponding to a give item. We refer to this second testbed system as *RSM.outnode*. We chose as baseline two of the most popular search engines on the market[3]: *Google* and *Bing*. One of the most prominent features of said search engines is in fact the ability to leverage the LOD cloud to improve search results, more specifically they can retrieve a neighborhood of items closely related to the search query given by the user. To obtain fair and generic search results i.e. not influenced by the recorded browsing history, prefer-

[3] http://www.alexa.com/.

ences, and location, Google and Bing search process was depersonalized to prevent the search engines from customizing the final result.

To assess the quality of our two alternative approaches we constructed a dataset of the top visited Wikipedia pages. As a reliable source of data we used the list of Wikipedia Popular Pages[4] that maintains a set of the most accessed 5000 articles on the English Wikipedia and it is updated weekly. For our data set we focused, for both English and Italian, on the most *stable articles* during the year 2015. We define the stable articles as the Wikipedia pages that constantly appear in every weekly version of that list throughout the year, and so receiving constant interest from the visitors of Wikipedia. A set of 1583 stable items were identified for the English language, and a set of 4361 for the Italian. Four evaluation datasets, two for English, and two for Italian, were built by randomly selecting from each language's stable articles list 100 items (used in experiment 1) and 25 items (used in experiment 2) upon which all of the four systems are able to retrieve related items.

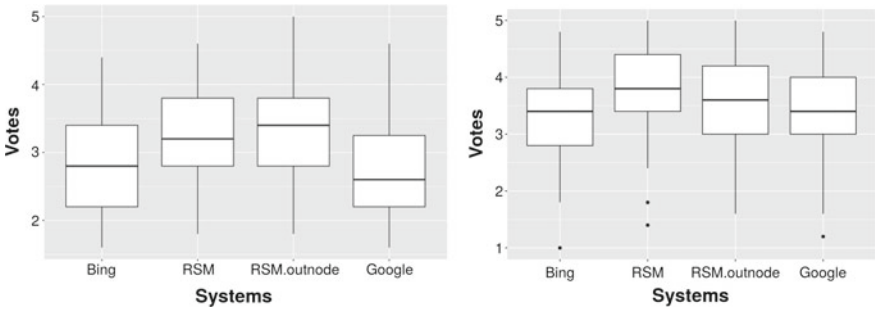
4.2 Overall Relevance Assessment

The goal of our first experiment was to assess which one of the four systems produces the overall best set of related items given one search key. To this extent, we considered datasets of 100 items. The crowdsourcing experiment was designed as follows: for each of the considered items a page was generated including the name of the item, a brief description, a picture, and a box including the results produced by the four systems i.e. four lists of five semantically related items. We decided to show only five results for two reasons: firstly both Bing and Google show at least five related items, which means that for some search queries no more than five items will be shown, secondly it is a known fact that users typically pay attention only to the top spots of search results lists, with the top five items attracting most of the attention.[5] To avoid cognitive bias, the names of the systems were not shown and the presentation order was randomized, so that the worker had no means of identifying the source of the presented item lists and couldn't be biased by personal preference or previous evaluations. The workers were then asked to rate the four item lists according to their perceived quality in terms of relatedness on a discrete scale from 1 to 5 where 1 meant total randomness and 5 that all presented items where perceived as strongly related. Each one of the 100 items in the data set was shown with the same related items lists to 5 distinct users and their judgements were averaged per system to mitigate subjectivity of judgement. The experiment was performed using the popular crowd sourcing platform *Crowdflower*[6] and iterated twice: once for the English

[4] https://en.wikipedia.org/wiki/User:West.andrew.g/Popular_pages.

[5] https://chitika.com/google-positioning-value.

[6] http://www.crowdflower.com/.



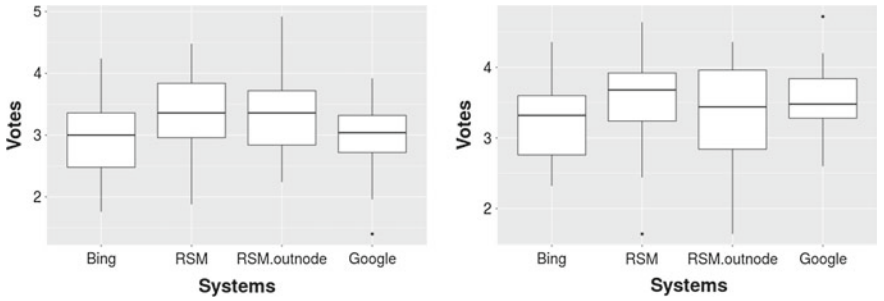
(A) English (B) Italian

Fig. 2 Experiment 1 distribution of worker's judgement

dataset and one for the Italian one. In the English iteration 32 users from 18 different countries were involved, with an average of 15.62 judgements per user. In the Italian iteration, instead, were involved 59 users from 8 countries, with an average of 8.47 judgements per user). The distribution of the worker's judgement is shown in Fig. 2.

4.3 Item by Item Relevance Assessment

The goal of our second experiment was to assess the perceived quality of each item included in the related items list. To this extent we considered datasets of 25 items. The experimental setup was similar to the previous experiment, using the same platform and displaying the same information about the target entity (i.e. title, description, and picture). Instead of four lists, this time the workers were shown a single list generated by one system only and were asked to rate each item in the list on a scale from 1 to 5 where 1 implied complete unrelatedness and 5 a very high perceived relatedness. The name of the system that generated the list was not shown to avoid bias. A hundred related items lists where therefore generated and human-rated item by item. Again, each item was judged by five distinct users to mitigate subjectivity of judgement. This second experiment was again iterated twice and involved by design substantially more workers to further abstract over subjective experience and thus obtain a more impartial judgement. In the end 146 workers from 38 countries were involved with an average of 3.42 judgements per user in the English experiment and 109 workers from 14 countries with an average of 4.59 judgements in the Italian one. In Fig. 3 the distribution of workers' judgements is shown.



(A) English (B) Italian

Fig. 3 Experiment 2 distribution of worker's judgement

5 Discussion

The data gathered with the experiments described in Sect. 4.1 provide some interesting insights on the effectiveness of the proposed technique.

5.1 Overall List Quality

The results of experiment one showed how our testbed systems RSM and RSM.outnode can achieve satisfactory performance in both the considered scenarios. In the English part of the experiment RSM and RSM.outnode achieved, on a scale from 1 to 5, respectively a 3.20 and 3.33 average perceived quality, while Google and Bing respectively 2.79 and 2.82. The statistical significance of the judgement distributions shown in Fig. 2 was evaluated as well showing how while there is a substantial difference between the perceived quality of our systems and the baseline ones (Bing and Google), between RSM and RSM.outnode there is no statistically significant difference. More specifically the Welch Two Sample t-test was used and produced the results shown in Table 1, where in the upper right half of the matrix are shown the p-values produced by the test, and in the lower left half the same values recalculated with the Benjamini & Hochberg correction for multiple hypotesis testing [1]. According to these results, Google's and Bing's related items lists are perceived almost as identical in terms of quality, while our testbed systems' outputs receive a significantly higher likely by the crowdsourced workers. Moreover, while RSM.outnode appears to achieve an higher perceived quality than RSM on average, the statistical significance analysis shows that such a difference is unlikely to be significant in the current experimental setting. In terms of overall perceived quality the neighbourhoods of related items to a given search key produced by RSM and RSM.outnode do not differ significantly in terms of perceived quality, but there is evidence that consistently outperform the benchmark systems offered by Google and Bing.

Table 1 Statistical significance of the difference between the considered systems over the English corpus. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction

	RSM	RSM.outnode	Google	Bing
RSM	–	0.1896	<0.0001	0.0001
RSM.outnode	0.2275	–	<0.0001	<0.0001
Google	<0.0001	<0.0001	–	0.6838
Bing	0.0003	<0.0001	0.6838	–

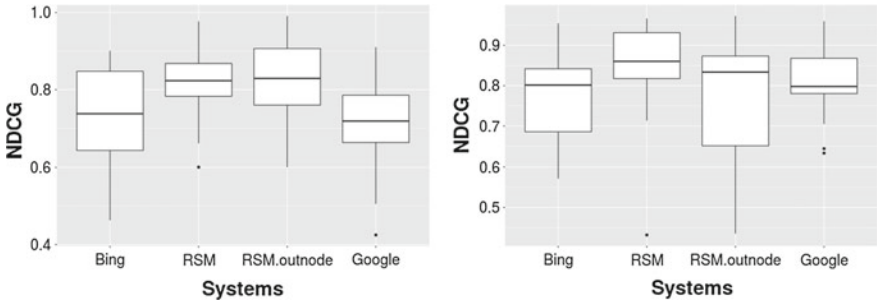
Table 2 Statistical significance of the difference between the considered systems over the Italian corpus. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction

	RSM	RSM.outnode	Google	Bing
RSM	–	0.0079	0.0013	<0.0001
RSM.outnode	0.0158	–	0.6835	0.0141
Google	0.0039	0.6835	–	0.0308
Bing	<0.0001	0.0125	0.0369	–

The Italian part of the experiment a similar outcome was observed, with two notable differences: expressed scores were substantially higher for all systems and in particular results produced by Google received a generally more favourable reception with respect to the English part of the experiment. While the former outcome may be ascribed to cultural factors, since the whole judgement distribution is skewed towards higher scores, the latter suggests that the localised versions of Google and Bing may differ in the used data or retrieval technique. As a matter of fact, the English Bing and Google received very similar judgements, see Table 1, and the provenance of the related items lists was unknown to workers to avoid confirmation bias, thus the significant difference sported in the Italian experiment, shown in Table 2, implies substantial differences between the English and the Italian versions of the two search engines. On the other hand, the RSM model appears the one producing the best received related items lists, while RSM.outnode and Google present no statistically significant difference. The statistically significant difference between the perceived quality of the lists generated by RSM and RSM.outnode in this setting can be ascribed to substantial reduction in the size of the training data. Overall, RSM is perceived as the best system, RSM.outnode and Google are on par, and Bing is perceived as the worst one.

5.2 Information Gain Analysis

The results of experiment two support the evidence provided by the previous one. In the English part of the experiment, items retrieved by RSM and RSM.outnode on average score a 3.41 out of 5 on perceived quality while Bing and Google stop at 2.93 out of 5. In the Italian part of the experiment, instead, items retrieved by RSM score an average of 3.6 out of 5, RSM.outnode and Google are tied around 3.5, and Bing scores around 3.4 on average. These numbers, however, provide little information being average values of perceived quality of item ranked in different positions. Looking at the whole distribution of judgements shown in Fig. 3, the high variance of the four distributions can be easily noticed. Such a variance can be justified by the fact that all items included in the generated lists are considered and rated. However, not all positions of a result list are equal to the extents of exploratory search. To address this issue we evaluated the Normalized Discounted Cumulative Gain (NDCG) of the four considered systems. NDCG is a metric commonly used in IR to assess a search engine's performance basing on the comparison between an ideal list of the most relevant retrievable items and the actual list produced by the evaluated system. Its core idea is that the higher the position of an item in the result list the more important the quality of that item should be in the quality evaluation of the system, therefore the presence of scarcely relevant items in the top spots tends to "punish" the evaluated system. The ideal list was computed by considering, for both parts of the experiment, for each of the 25 search keys, all the items retrieved by the four systems, picking the five ones that on average received the highest user ratings and ordering them in descending average rating order. The distribution of the NDCG values scored by the four considered systems over the search queries included in the data sets is shown in Fig. 4 and its detailed statistics are presented in Tables 3 and 4. These results support the evidence brought by the first experiment as well, with RSM and RSM.outnode providing consistently results perceived as more relevant than the ones brought by Google's and Bing's tools in the English part of the experiment. Again, there is no statistically significant difference in the average perceived quality between RSM and RSM.outnode (p-value = 0.68) and between Google and Bing as well (p-value = 0.88). On the other hand, the statistical significance between RSM and Google, RSM and Bing, RSM.outnode and Google, and RSM.outnode and Bing is high with p-values below 0.0001. Finally, the NDCG analysis shows how, despite scoring being on average on par with its RSM.outnode counterpart, the RSM system has the smallest variance in the perceived relevance of its results, implying that it is less likely to produce results perceived as poor on a single-try basis. In the Italian part of the experiment, instead, RSM achieves substantially higher nDCG scores than its RSM.outnode counterpart, which, again, presents a vary large nDCG score distribution and, on average, performs slightly worse than Google's related items search, though its median nDCG value is higher than Google's. Like in the previous experiment, the RSM model appears to be able to cope better with changes in training data.



(A) English (B) Italian

Fig. 4 NDCG values distribution evaluated on the results of experiment 2

Table 3 Distribution statistics on NDCG evaluation—English

	Bing	RSM	RSM.outnode	Google
Minimum	0.4629	0.6009	0.6006	0.4250
1st Quartile	0.6423	0.7829	0.7601	0.6631
Median	0.7376	0.8232	0.8293	0.7186
Mean	0.7247	0.8113	0.8226	0.7196
3rd Quartile	0.8475	0.8678	0.9066	0.7855
Maximum	0.9010	0.9771	0.9910	0.9102

Table 4 Distribution statistics on NDCG evaluation—Italian

	Bing	RSM	RSM.outnode	Google
Minimum	0.5714	0.4319	0.4352	0.6329
1st Quartile	0.6859	0.8177	0.6511	0.7804
Median	0.8015	0.8602	0.8338	0.7980
Mean	0.7793	0.8493	0.7664	0.8121
3rd Quartile	0.8418	0.9313	0.8733	0.8677
Maximum	0.9546	0.9664	0.9726	0.9598

Finally, it is important to stress how the MinHash optimisation allowed us to move the complexity of a pairwise similarity measurement from linear to constant. This means that without the said optimisation it would be computationally demanding to retrieve items semantically related to one with a lot of connections. Consider for instance the Wikipedia article about Barack Obama which, at the time this article being written, contained over 250 links was referenced over 9900 times by other Wikipedia articles: without MinHash it takes over 300 s on our test machine[7] to generate a list of semantically related items, while with that optimisation it takes less

[7] An Intel I7 with eight cores and 32 GB RAM.

than a second on the same machine. Moreover, the constant complexity of MinHash allows it to seamlessly scale up to larger knowledge bases. While our approach allows this optimisation to be made retaining quality results, other metrics, such as the ones presented in [6, 23], do not.

6 Conclusions

Our evaluation provided concrete evidence that our approach is able to achieve results consistently perceived as satisfactory by the crowdsourced workers. In particular, the referential model built upon considering references rather than outgoing links appears to be more robust as the training data and the users change. The referential hypothesis thus allowed us to build a sound VSM that captures semantic relatedness among the considered items, and the usage of the Jaccard index and the MinHash optimisation allowed us to handle the over 8 million items included in Wikipedia with ease. Providing semantic tools that can efficiently scale up to the large volumes of data involved in nowadays information access applications such as personalised information retrieval and personalised recommendation, is in our opinion a critical step towards fully accomplishing the potential of the Web of data. It is well known that the graph nature of the LOD cloud implies high computational costs when exploration and reasoning tasks must be performed and many current state of the art algorithms involve extensive graph traversing. For instance, it took over 2,400 min to the authors of [16] to train a state of the art semantics-based recommender system on a data set such as *DBbook*[8] which is relatively small when compared to real-world scenarios that may include millions of items and users. Though training is typically a batch-time operation, an excessive complexity may discourage its field usage since in Adaptive Personalisation applications it typically needs to be frequently repeated because it is likely to users to regularly update their preferences, new items to be included and new users to register.

References

1. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. Royal Statist. Soc. Ser. B Methodol. **57**(1), 289–300 (1995)
2. Broder, A.Z.: On the resemblance and containment of documents. In: Proceedings of Compression and Complexity of Sequences (SEQUENCES'97), pp. 21–29. IEEE, June 1997
3. Alexander, B., Graeme, H.: Semantic distance in wordnet: an experimental, application-oriented evaluation of five measures. In: Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 29–34 (2001)
4. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Comput. Linguist. **32**(1), 13–47 (2006)

[8] http://challenges.2014.eswc-conferences.org/index.php/RecSys#DATASET.

5. Rudi, L.C., Paul, M.B.V.: The google similarity distance. IEEE Trans. Knowled. Data Eng. **19**(3), 370–383 (2007)
6. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJcAI **7**, 1606–1611 (2007)
7. Risto, G., Warner ten K., Zharko, A., Frank Van H.: Using google distance to weight approximate ontology matches. In: The 16th International Conference on World Wide Web, pp. 767–776. ACM, (2007)
8. Sebastien, H., Sylvie, R., Stefan, J., Jacky, M.: Semantic similarity from natural language and ontology analysis. Synth. Lect. Human Lang. Technol. **8**(1), 1–254 (2015)
9. Harris, Z.: Distributional structure. Word **10**(23), 146–162 (1954)
10. Tin, H., Kiem, H., Loc, Do, Huong, T., Hiep, L., Susan, G.: Scientific publication recommendations based on collaborative citation networks. In: Collaboration Technologies and Systems (CTS), 2012 International Conference on, pp. 316–321. IEEE, (2012)
11. Jaccard, P.: Lois de distribution florale. Bulletin de la Socíeté Vaudoise des Sciences Naturelles **38**, 67–130 (1902)
12. Lillian, L.: Measures of distributional similarity. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL), (199)
13. Leskovec, J., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, (2014)
14. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Trans. Assoc. Comput. Linguist. **3**, 211–225 (2015)
15. Christopher, D.M., Prabhakar, R., Hinrich, S.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
16. Cataldo, M., Pasquale, L., Pierpaolo, B., Marco de G., Giovanni, S.: Semantics-aware graph-based recommender systems exploiting linked open data. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, pp. 229–237. ACM, (2016)
17. Novak, J.D.: Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. Taylor & Francis, London, United Kingdom (2010)
18. Mohammad, T.P. Roberto, N.: From senses to texts: an all-in-one graph-based approach for measuring semantic similarity. Artific. Intell. **228**, 95–128 (2015)
19. Rodríguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. IEEE Trans. Knowled. Data Eng. **15**(2), 442–456 (2003)
20. Turney, Peter D.: Pantel, Patrick: from frequency to meaning: vector space models of semantics. J. Artif. Int. Res. **37**(1), 141–188 (2010)
21. Jingdong, W., Heng, T.S., Jingkuan, S., Jianqiu, J.: Hashing for similarity search: a survey. arXiv:1408.2927, (2014)
22. Weeds, Julie: Weir, D.: Co-occurrence retrieval: a flexible framework for lexical distributional similarity. Comput. Linguist. **31**(4), 439–475 (2005)
23. Ian, W., David, M.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, AAAI Press, Chicago, USA, pp. 25–30(2008)