Getting Creative with Semantic Similarity

Ching-Yun Chang
University of Cambridge
Computer Laboratory
15 JJ Thomson Avenue
Cambridge, UK CB3 0FD
cyc30@cam.ac.uk

Stephen Clark
University of Cambridge
Computer Laboratory
15 JJ Thomson Avenue
Cambridge, UK CB3 0FD
sc609@cam.ac.uk

Brian Harrington
University of Toronto Scarborough
Computer and Mathematical Sciences
1265 Military Trail, Toronto
Ontario, Canada M1C 1A4
brian@brianharrington.net

***Abstract*—This paper investigates how graph-based representations of entities and concepts can be used to infer semantic similarity and relatedness, and, more speculatively, how these can be used to infer novel associations as part of a creative process. We show how personalised PageRank on a co-occurrence graph can obtain competitive scores on a standard semantic similarity task, as well as being used to discover interesting and surprising links between entities. We hypothesise that such links could form the first stage in a creative ideation process.**

I. Introduction

This paper investigates how graph-based representations of enities and concepts can be used to infer semantic similarity ratings, and how these can then be used to infer novel associations as part of a creative process. In terms of inferring semantic similarity, we are following a recent line of work in which graph-traversal algorithms, such as personalised PageRank and spreading activation, are run over manually or automatically constructed semantic networks [1], [2]. In terms of creatively exploiting semantic similarity, our work fits in the broader research programme known as bisociation knowledge discovery [3], and is also related more generally to the emerging field of computational creativity [4], [5], [6].

The novel aspect of this short paper is a suggestion for how semantic networks can be exploited for the process of *ideation* — the creative process of generating new ideas. These ideas could then be used in a wider computational creativity system, for example one which creates visual collages [7], generates poetry [8], or produces puns or jokes [9].

Bisociation was proposed by Koestler as a general process for combining two previously unrelated ideas or thoughts into a new idea [10], and one which is claimed to be at the heart of many inventions and discoveries. Much of the existing research in bisociation knowledge discovery [3] has integrated information from various domains into a single information network in order to explore potentially interesting links between (apparently) unrelated concepts. For example, [11] constructed a graph from clinical notes in which nodes are symptoms and edges represent co-occurrence relationships between symptoms. A random walk algorithm [12] was then applied to the graph in order to try and link symptoms to a disease.

Harrington proposes a similar idea for identifying novel relationships between biomedical entities such as genes and proteins, based on an automatically-constructed network from parsed biomedical research papers [13]. The novelty arises from finding entities which are closely related in the network — as measured by a spreading activation metric — but which crucially have not occurred in the same document. [13] contains a non-biomedical example in which Richard Socarides is discovered to be closely related to Bill Clinton (Socarides was a White House advisor to Clinton), even though Clinton and Socarides were never mentioned in the same documents on which the network was based.

In this paper we show how Harrington's idea can be used to find surprising and creative links between entities. For example, we discover a strong and unexpected link between Michael Jackson, the singer, and Martina Navratilova, the tennis player. The link occurs because both are related to the *Billie Jean* node in the graph — Jackson because of the song, and Navratilova because of the tennis player Billie Jean King.

II. Network Construction

The large-scale semantic networks are built from sentences in Wikipedia. The key decisions in defining the networks are what the nodes and edges represent, and how to weight the edges. Here we use content words and named entities as nodes, and link nodes with edges if the corresponding words have appeared in the same context. We experimented with a variety of context definitions, including using typed dependency parser output as in [13], but found that a simple definition of context, in which words appearing in the same sentence are linked with an edge, performed best in the semantic similarity experiments described below.

The Wikipedia dump was processed using the C&C tools [14], together with the Wikipedia named entity model from [15]. Nodes are either single words pos-tagged with noun, verb, adjective, or adverb tags, or single or multi-word units tagged with a named entity label (`PERSON`, `ORGANISATION`, `LOCATION` or `MISC`). Only nodes with a frequency of at least 100 across the corpus are retained in the network, which resulted in a total of 150,298 nodes.[1] Formally it is covenient to think of the network as a directed weighted graph, with two

[1]This threshold was chosen to meet the competing requirements of having nodes which are well-represented in the data; a managable graph in terms of size; but still having a large-scale graph with many nodes.

weighted edges — one for each direction — between nodes which co-occur together in at least one sentence.

We experimented with a variety of edge weights, all based on the frequency of co-occurrence between nodes. All the edge weighting mechansisms are based on standard association metrics used in the lexical semantics literature [16]: mutual information (MI), log-likelihood ratio (LLR), term frequency-inverse collection frequency (TF-ICF) and chi-square (CHI). All these association measures are calculated using the following statistics: the number of times that the two nodes co-occur; the number of times that the source node co-occurs with other nodes; and the total number of co-occurrences of any nodes across the whole corpus.

Finally, the weights for all those edges having the same source node are normalised to sum to 1, so that personalised PageRank can be run on the graph. The intuition behind the weighting mechanism is that we would like the PageRank random surfer to prefer moving along edges which relate highly associated nodes.

A. Personalised PageRank

PageRank, roughly speaking, measures the probability of a "random surfer" being on a particular node in the graph [17]. Usually the random surfer chooses an edge linked to the current node at random — according to the normalised weights on each edge — and moves to an adjacent node. However, there is a parameter in the PageRank algorithm which allows the surfer, with some probability, to make a random jump to any other node in the network. PageRank can be *personalised* to a target node by putting all the random-jump probability mass on that node, so that when the surfer performs the random jump, it is always to that node. From another perspective, PageRank computes the global importance of a particular node in the network, whereas the notion of importance for a particular node in personalised PageRank is defined relative to a target node. We use personalised PageRank to measure the importance, interpreted as semantic relatedness in our application, of every other node in the network, relative to a target node.

Personalised PageRank values are obtained by computing the stationary distribution of a Markov chain, with a parameter β denoting the probability of moving to the state in which the random surfer is on the target node. Let G be the graph described above with N nodes, and $\vec{v}^{(t)} \in \mathbb{R}^N$ denote the probability distribution over the N nodes after t steps of the random walk. $\vec{v}^{(0)}$ is initialised so that all the probability mass is on the target node. The transition probability matrix $M \in \mathbb{R}^{N \times N}$ is defined such that the (i, j) entry of M is equal to the probability of moving from node j to node i; that is, column j contains node j's outgoing probability distribution over the N nodes. The stationary distribution $\vec{v}^{(t)}$ is computed via the following iterative algorithm:

$$\vec{v}^{(t)} = \beta\vec{v}^{(0)} + (1 - \beta)M\vec{v}^{(t-1)}$$

Once the iterative algorithm has converged, we obtain a vector of normalised values, relative to the target node, over all nodes in the network. This is the semantic relatedness vector as computed by personalised PageRank, given the particular context and weighting definitions we use.

III. Semantic Relatedness Evaluation

To allow comparison with previous results on semantic similarity or relatedness tasks, we use the WordSimilarity-353 (WS-353) dataset [18], which has been widely used in the literature [19], [20], [1], [21]. It consists of 353 word-pairs, each of which was evaluated by up to 17 human judges who were instructed to rate the *relatedness* between the two words in each pair on a scale of 0 (totally unrelated words) to 10 (very much related or identical words).[2] For each word-pair the mean score is assigned as the gold standard. For example, the mean human relatedness score for the pair (*midday*, *noon*) is 9.29, while the score for the pair (*king*, *cabbage*) is 0.23.

To automatically calculate the relatedness between two words, e.g. *midday* and *noon*, we first calculate two relatedness vectors by running personalised PageRank twice on the complete Wikipedia graph described earlier, with each word as the target node. Then the two relatedness vectors are compared. So for the example pair, we first get a relatedness vector for *midday*, which contains PageRank scores for every node in the graph relative to *midday*, and then calculate the corresponding vector for *noon*. Finally these two vectors are compared.

We experimented with a variety of metrics to compare the vectors [19], including COSINE, Euclidean distance and KL-divergence. In fact, we found that a simpler method which only uses the PageRank values of the two words in each other's relatedness vectors performed best. For this simple method, suppose the two nodes being compared are *midday* and *noon*. We take the PageRank value of *midday* from the vector calculated with *noon* as the personalised node; the PageRank value of *noon* from the vector calculated with *midday* as the personalised node; and then take the average of the two values.

The resulting similarity scores for each pair in the dataset are sorted and the pairs ranked accordingly. The same procedure is applied to the human scores to obtain the gold standard ranks. Then the two ranked lists are compared using Spearman's rank correlation coefficient [23] which gives a value between -1 and 1, with 1 indicating a perfect Spearman correlation and 0 no correlation.

Table I shows the Spearman correlation results obtained with the different edge weighting methods described in Section II, and the simple and cosine distances described above. The table also demonstrates the effect of the value of the random-jump probability β. It is clear that a higher β value generally outperforms a lower β value. This is not surprising since, the higher the β value is, the more personalised the relatedness vector is for a target node. The best result for the simple method is comparable to [24] who report a highly competitive correlation of 0.75 on a Wikipedia-based graph

[2]The distinction is often made in the literature between semantic *relatedness* and semantic *similarity* [22]. For example, *bank* and *trust company* are similar, whereas *car* and *wheel* are related [22]. In this paper we do not make a distinction between the two.

Weights	β	SIMPLE	COSINE
FREQ	10^{-4}	0.51	0.19
	0.15	0.58	0.20
	0.50	0.64	0.19
	0.99	0.67	0.20
MI	10^{-4}	-0.12	0.15
	0.15	0.12	0.03
	0.50	0.53	0.05
	0.99	0.38	0.20
LLR	10^{-4}	-0.05	0.06
	0.15	0.56	0.14
	0.50	0.67	0.13
	0.99	0.70	0.14
TF-ICF	10^{-4}	0.71	0
	0.15	0.74	0.25
	0.50	0.74	0.46
	0.99	0.73	0.56
CHI	10^{-4}	0.21	0.16
	0.15	0.73	0.36
	0.50	0.72	0.41
	0.99	0.70	0.41

TABLE I
SPEARMAN CORRELATION RESULTS ON WS-353

(although using different techniques to those proposed here, in particular exploiting the encyclopaedic nature of Wikipedia).

IV. BISOCIATION EXTRACTION

The previous section showed that graphs built from sentences in Wikipedia, together with a simple definition of context based on sentence boundaries, can lead to high-quality semantic similarity ratings through the use of personalised PageRank. In this section we show that the same techniques can be adapted to discover some interesting and creative links between entities.

The nodes we focus on as target and related nodes are those which are labelled with named entity tags. Consider the example of finding named entities related to *Michael_Jackson*. First personalised PageRank is run on the complete weighted co-occurrence graph built from Wikipedia (including all content nodes, not just named entities), with *Michael_Jackson* as the target node. The nodes with the highest PageRank value relative to Michael Jackson are as expected, i.e. those named entities which occur frequently in the same sentences as the phrase *Michael Jackson*, but occur less frequently with other nodes (depending on the particular weighting scheme used). Hence *Janet_Jackson* receives a high PageRank value relative to *Michael_Jackson*.

Discovering that Michael Jackson is closely related to his sister is not particularly interesting, however. Hence the nodes that we focus on are those which are *at least two edges away from the target node*. The top row of Table II shows some interesting examples for the *Michael_Jackson* example. The number in brackets is the rank, according to the PageRank value, ignoring all those nodes which are only one edge away from the target node. The node in brackets is the node which results in the shortest path between the target and related node, where the score of a path is the sum of the negative log probabilities on the edges (which is equivalent to preferring high probability edges, i.e. ones which "pass on" mass during the PageRank calculation).

Michael Jackson is found to be related to the tennis player Martina Navratilova through the *Billie_Jean* node. This connection arises because Martina Navratilova is closely related and directly connected to another famous U.S. tennis player, Billie Jean King, and Michael Jackson is closely related and directly connected to his song Billie Jean. Note that we are not claiming that Michael Jackson is semantically similar or semantically related to Martina Naratilova (or at least not directly). On the contrary, the claim is that interesting links between entities can be found by exploiting direct semantic similarity ratings between nodes, with an intermediate node (which is semantically related to both entities) exposing the link.

For the other Michael Jackson examples, Michael Jackson was famous for a particular kind of walk he often did during his performances, called a *moonwalk*, and Neil Armstrong is famous as the first man to walk on the moon. *Thriller* is the name of another Michael Jackson song, and the Library of Congress contains many types of thriller novels.

All of these examples are interesting because they demonstrate a feature of language which is often exploited in creative processes, namely ambiguity. The Michael Jackson–Martina Navratilova example is particularly pertinent in this regard. Here the *Billie_Jean* node is ambiguous between the song and tennis player, and in fact with a more discerning named entity tagger, the two instances of *Billie Jean* would have been tagged differently. Hence the ideation process is also exploiting to some extent weaknesses in the underlying language technology. To what extent this is a bug or a feature is an important question for future work, given the importance of ambiguity in creative processes.

Another interesting link with Michael Jackson occurs with a different target node, namely the Catholic Church. Here the high-profile stories surrounding both Michael Jackson and the Catholic Church regarding child abuse are responsible for the link (with the noun *molestation* providing the intermediate node leading to the highest-scoring path between the two nodes).

For the *James Bond* target node, Rachel Weisz is a famous actress married to the actor who currently plays James Bond, Daniel Craig. Edinburgh is the home of Fettes College where the fictional character studied. And Heathcliff is a famous fictional character who has been played by Timothy Dalton, who also played James Bond.

The reader is left to study the table and consider what the link may be between the nodes in the remaining examples. A few remarks are in order here. First, the examples have been cherry-picked by the authors. We consider this acceptable for two reasons. One is that a possible use for this application is to help workers in the so-called creative industries — e.g. advertising — find interesting links between entities. Hence we envisage some human involvement in sifting through the list to find the interesting cases. For example, a satirical artist working for a newspaper could quickly look through the

Target entity	Related node with ranking and intermediate node in brackets
Michael_Jackson	*Martina_Navratilova* (2, *Billie_Jean*); *Neil_Armstrong* (39, *moonwalk*); *Library_of_Congress* (172, *thriller*)
Catholic_Church	*Stephen_Jay_Gould* (1, *magisterium*); *ZFC* (50, *cardinal*); *Michael_Jackson* (601, *molestation*)
James_Bond	*Rachel_Weisz* (9, *Daniel_Craig*); *Edinburgh* (106, *Fettes_College*); *Heathcliff* (182, *Timothy_Dalton*)
Burger King	*homer* (1, *Whopper*); *Spurlock* (10, *fast-food*); *Peter_Pan* (80, *Wendy*)

TABLE II
HIGHLY-RELATED ENTITY PAIRS WITH SURPRISING LINKS

ranking and find the Michael Jackson and Catholic Church example.

A second reason is that, even in a fully automatic system, the process described here would only be the first stage in the ideation process, and a second stage would need some knowledge of what makes a link interesting in order to filter the useless examples. As Boden makes clear [6], the sort of *combinational creativity* being attempted here is relatively easy for a computer: picking two ideas and placing them next to each other is not a difficult program to write. In this paper we have attempted to take the trivial connecting of ideas a stage further, by using a large-scale, automatically created semantic resource together with a graph-based ranking mechanism. However, it is clear that for a computer to even show indications of appearing creative it would need a sophisticated filtering mechanism.

Hence the second stage would need a novelty-ranking mechanism based on what is potentially interesting (given the application in hand), but could also take into account other clutural factors such as timeliness, perhaps based on what is featuring in the current news or trending on Twitter. This filtering stage is consistent with the ideation process in humans, in which many useless ideas are typically generated, but only a few are chosen.

The second point is that the authors acknowledge that providing a list of manually chosen examples does not constitute a strong evaluation. However, evaluating creative processes is problematic and beyond the scope of this short paper. What we are offering here is a proof-of-concept and the suggestion that the idea is worth pursuing, as part of the new emerging fields of computational creativity and bisociation extraction.

ACKNOWLEDGMENTS

The first two authors were supported by a Google Research Award: Knowledge Extraction and Discovery from Large-Scale Entity-Relationship Graphs.

REFERENCES

[1] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "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*, ser. NAACL '09, Boulder, Colorado, 2009, pp. 19–27.

[2] B. Harrington, "A semantic network approach to measuring relatedness," in *Proceedings of COLING 2010*, Beijing, China, 2010.

[3] M. R. Berthold, Ed., *Bisociative Knowledge Discovery*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7250.

[4] S. Colton and G. A. Wiggins, "Computational creativity: The final frontier," in *Proceedings of the 20th European Conference on Artificial Intelligence*, Montpellier, France, 2012.

[5] T. Veale, *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity*. London: Bloomsbury Academic, 2012.

[6] M. A. Boden, *The Creative Mind – myths and mechanisms (2nd Edition)*. Routledge, 2004.

[7] A. Krzeczkowska, J. El-Hage, S. Colton, and S. Clark, "Automated collage generation - with intent," in *Proceedings of the 1st International Conference on Computational Creativity*, Lisbon, Portugal, 2010.

[8] S. Colton, J. Goodwin, and T. Veale, "Full-face poetry generation," in *Proceedings of the 3rd International Conference on Computational Creativity*, Dublin, Ireland, 2012.

[9] G. Ritchie, "The jape riddle generator: technical specification," University of Edinburgh, Technical Report EDI-INF-RR-0158, 2003.

[10] A. Koestler, *The Act of Creation*. Hutchinson, 1964.

[11] P. Sondhi, J. Sun, H. Tong, and C. Zhai, "SympGraph: a framework for mining clinical notes through symptom relation graphs," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '12. Beijing, China: ACM, 2012, pp. 1167–1175.

[12] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[13] B. Harrington, "Discovering novel biomedical relations using ASKNet semantic networks," in *Proceedings of the 4th International Symposium On Applied Sciences In Biomedical And Communication Technologies (ISABEL 2011)*, Barcelona, Spain, 2011.

[14] J. R. Curran, S. Clark, and J. Bos, "Linguistically motivated large-scale NLP with C&C and Boxer," in *Proceedings of the ACL 2007 Demonstrations*, Prague, Czech Republic, 26 June 2007, pp. 33–36.

[15] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, "Learning multilingual named entity recognition from Wikipedia," *Artificial Intelligence*, vol. 194, no. 0, pp. 151–175, 2013.

[16] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[17] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[18] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: the concept revisited," *ACM Transactions on Information Systems*, vol. 20, no. 1, pp. 116–131, 2002.

[19] J. R. Curran, "From distributional to semantic similarity," Ph.D. dissertation, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, 2003.

[20] M. Strube and S. P. Ponzetto, "WikiRelate! computing semantic relatedness using Wikipedia," in *proceedings of the 21st national conference on Artificial intelligence*, ser. AAAI'06, Boston, Massachusetts, 2006, pp. 1419–1424.

[21] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th international conference on World Wide Web*, ser. WWW '11, Hyderabad, India, 2011, pp. 337–346.

[22] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of semantic distance," *Computational Linguistics*, vol. 32, pp. 13 – 47, March 2006.

[23] C. Spearman, "Correlation calculated from faulty data," *British Journal of Psychology*, vol. 3, no. 3, pp. 271–295, 1910.

[24] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *Proceedings of the 20th international joint conference on Artifical intelligence*, ser. IJCAI'07, Hyderabad, India, 2007, pp. 1606–1611.