A folksonomy-based approach for profiling human perception on word similarity

GuanI Wu
Department of Statistics,
University of California, Los Angeles
guani@g.ucla.edu

Ker-Chau Li
Department of Statistics,
University of California, Los Angeles
ISS, Academia Sinica
kcli@stat.sinica.edu.tw

Abstract

Automatic assessment of word similarity has long been considered as one important challenge in the development of Artificial Intelligence. People often have a big disagreement on how similar a pair of words is. Yet most word similarity prediction methods, taking either the knowledge-based approach or the corpus-based approach, only attempt to estimate an average score of human raters. The distribution aspect of similarity for each word-pair has been methodologically neglected, thus limiting their downstream applications in Natural Language Processing. Here, utilizing the category information of Wikipedia, we present a method to model similarity between two words as a probability distribution. Our method leverages unique features of folksonomy. The success of our method in describing the diversity of human perception on word similarity is evaluated against the rater dataset WordSim-353. Our method can be extended to compare documents.

1 Introduction

Making machine understand human language is one of the ultimate goals in the development of Artificial Intelligence (Christopher D. Manning, 2015). In order to reach the goal, many different Natural Language Processing (NLP) tasks were designed. Among them, one of the fundamental upstream tasks is to automatically assess similarities between words. The performance of this task has directly impacts on many downstream NLP applications such as Question Answering, Information Retrieval, Topic Modeling, and Text Clustering (Sandhya and Govardhan, 2012; Nathawitharana et al., 2016; Wei et al., 2015), etc.

Methods automatically assessing word similarity generally fall into two categories, knowledge-based and corpus-based approaches (Harispe et al., 2015). The corpus-based approach was founded on the maxim “You should know a word by the companies it keeps (Firt, J. R., 1957), which has shown remarkable performance on different word-similarity tests. Landauer et al. proposed Latent Semantic Analysis (LSA) that employs singular value decomposition to generate vectors as word representations (Thomas K Landauer et al., 1998). Since then, many methods were proposed to generate word vectors. Bengio et al. published a series of papers using neural network techniques (Yoshua Bengio et al., 2003). The team of Tomas Mikolov proposed the continuous bag of words (CBOW) and skip grams (also known as Word2vec) (Tomas Mikolov et al., 2013) and Jeffrey et al. proposed GloVe (Pennington et al., 2014). These methods need to be fed with a large corpus to train models in order to generate word vectors. To obtain a similarity score between two words, the dot product of the two word vectors is computed.

Instead of the dependence on which corpus to use, the knowledge-based approach requires a pre-existing knowledge base. WordNet is the most common knowledge base employed by the majority of methods developed in this realm. WordNet collects over 150,000 English words, and organizes them into cognitive synonyms (synsets). These synsets are connected through conceptual, semantic and lexical relations such as hyponyms, hypernyms, meronyms, holonyms (George A. Miller, 1995). Wu and Palmer proposed a method that exploited ontology/taxonomy to compute similarity scores based on Least Common Subsumer (LCS) (Zhibiao Wu and Martha Palmer, 1994). Many methods based on LCS, known as the edge-counting-based approach, were proposed (T. Slimani et al., 2006; Yuhua Li et al., 2003; Hadj Taieb et al., 2014). Another type of knowledge base approach used features of words to assess the similarities (Amos Tversky, 1977; Andrea Rodriguez and Max J Egenhofer, 2003; Euripides G.M. Petrakis et al., 2006).

The performance of computed similarity has to be evaluated against human raters, but human raters often display considerable disagreement in assigning similarity scores. As an example, see Figure 1 for the distribution of 16 raters' scores assigned to the pair of *life* and *lesson* from **WordSim-353** (Finkelstein et al., 2002). Such rating disagreements are quite common. However, most word-similarity methodologies attempt to estimate only the "average" score of human rating. The distribution aspect has been methodologically neglected, thus limiting their downstream applications in NLP.

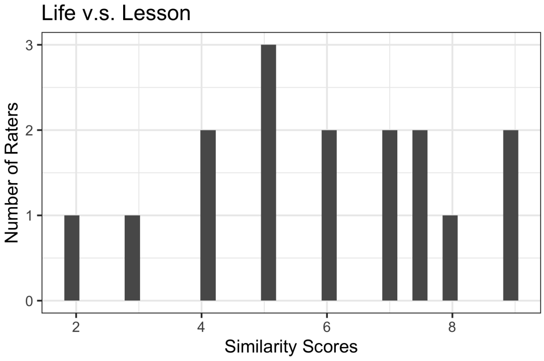
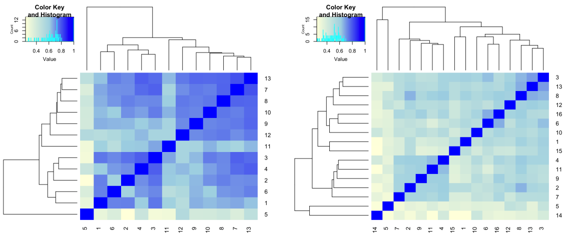


Figure 1: The histogram of similarity scores assigned by 16 raters to the pair of *life* and *lesson*.

2 Rater Disagreement on Word-Similarity

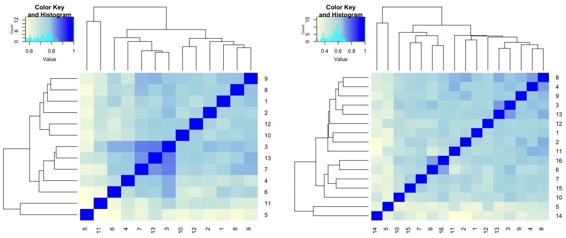
WordSim-353 is composed of two datasets: **WordSim-353.1**, a list of 153 word-pairs rated by 13 persons, and **WordSim-353.2**, a list of 200 word-pairs rated by 16 persons. We computed the Pearson correlation coefficient and the weighted Cohen's kappa coefficient for the similarity scores between any two raters. The results are shown in Figure 2 and Figure 3 after we ordered raters by hierarchical clustering. Rater disagreement on word-similarity is evident.

The important message we like to deliver is two-fold. First, the computer-imputed single similarity score has grossly simplified the human behavior. Second, using average rater score to evaluate the performance of different word-similarity prediction algorithms is itself a problematic evaluation approach.



(a) WordSim-353.1 (b) WordSim-353.2

Figure 2: Weighted Cohen's kappa coefficient matrices for WordSim-353.1 and WordSim-353.2.



(a) WordSim-353.1 (b) WordSim-353.2

Figure 3: Pearson correlation matrices for WordSim-353.1 and WordSim-353.2.

3 Leveraging Folksonomy for Distribution Quantification of Word Similarity

To reflect the more realistic human behaviors, we propose that in lieu of assigning a single similarity score, a better computer task would be to assign a probability distribution to each word-pair, $(p_0, p_1, \ldots, p_d, \ldots, p_\delta)$, where p_d denotes the probability of similarity score d, and δ is the highest allowable score. To evaluate the performance of a computer algorithm, we should employ common statistical criteria that are designed for the distribution against distribution comparison.

3.1 Category Information of Wikipedia

Wikipedia organizes the categories of articles via folksonomy, which is a collaborative tagging system allowing users to tag articles with multiple category notions (Aniket Kittur and Ed H. Chi, Bongwon Suh, 2009). Links between categories do not impose any specification on relations such as *is-a*, *is-part-of*, *is-an-example-of*, etc. Figure 4 illustrated how Wikipedia category is organized into a Directed Acyclic Graph (DAG). It is typical to find multiple roots linking to the title of an article.

In contrast to the traditional centralized classification, folksonomy may directly reflect the diversity of article contributors in their personal styles of vocabulary management, which in turn are influenced by a variety of factors including cultural, social or personal bias. At this writing, about 70,000 editors—from expert scholars to casual readers—regularly edit Wikipedia. (March 2, 2019 https://en.wikipedia.org/wiki/Wikipedia:About)

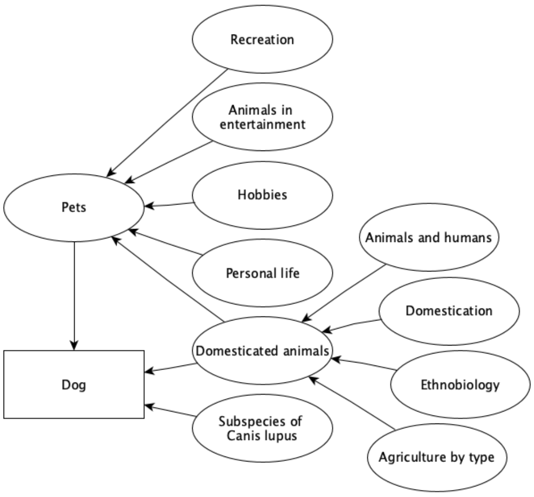


Figure 4: An example of Wikipedia category structure, where rectangle indicates a title of an article, and ellipses are categories. The graph is drawn based on the data downloaded from https://wiki.dbpedia.org/data-set-36.

3.2 Distribution Quantification of Word-Similarity

We propose a method to assign a probability distribution to a pair of words (W_1, W_2). First, we find the set of conceptual paths $X = \{X_1, \dots, X_N\}$ linking to W_1, and also find the set of conceptual paths $Y = \{Y_1, \dots, Y_M\}$ linking to W_2. We delete paths in X that are disconnected from any path in Y, and vice versa. We then compute a similarity score c_{ij} for each path pair (X_i, Y_j) to generate a matrix as shown in Table 1. The probability of similarity score d, denoted by p_d, is set to be the proportion of path pairs with $c_{ij} = d$.

We propose Equation 1 to calculate the similarity score for (X_i, Y_j).

$$sim(C_i, C_j) = 1 - \frac{(K_i + K_j)}{L_i + L_j} \propto L_i + L_j - K_i - K_j \quad (1)$$

As illustrated by Figure 5, L_i is the number of nodes on the path from C_i to its root node R_i, and L_j is the number of nodes on the path from C_j to its root node R_j. K_i is the number of nodes on the path from C_i to C_k, and K_j is the number of nodes on the path from C_j to C_k.

Y \ X	X_1	X_2	$\dots$	X_N
Y_1	c_{11}	c_{12}	$\dots$	c_{1N}
Y_2	c_{21}	c_{22}	$\dots$	c_{2N}
$\vdots$	$\dots$	$\dots$	$\ddots$	$\vdots$
Y_M	c_{M1}	c_{M2}	$\dots$	c_{MN}

Table 1: Matrix of Similarity Degrees Between Sets of Conceptual Paths.

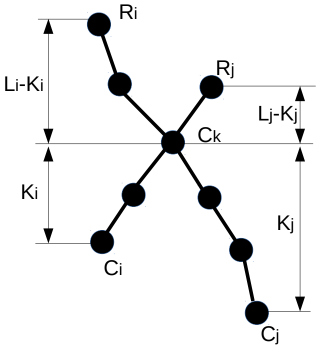


Figure 5: Calculating similarity between two conceptual paths via node counting.

In our implementation, we set L_i and L_j as constants and let $L_i = L_j = L$. There are two reasons. First, nodes that are too far away from C_i, C_j are often un-informative. Second, due to the large number of conceptual paths in X and Y, we must alleviate computational complexity. This leads to

$$c_{ij} = 2L - K_i - K_j \quad (2)$$

3.3 Implementation

Since there are over one million categories contained in Wikipedia, it would be a challenge to collect data directly from Wikipedia. Fortunately, DBpedia has collected and organized Wikipedia data in a way easier for us to use (Auer et al., 2007). We downloaded two datasets, *article-categories* and *skos-categories*; the former keeps the links between articles and categories, and the latter stores links between categories. Since the downloaded databases are stored in Triple-store format, *subject-predicate-object*, we set up Apache Jena Fuseki as an in-house SPARQL server for access by our main program. Figure

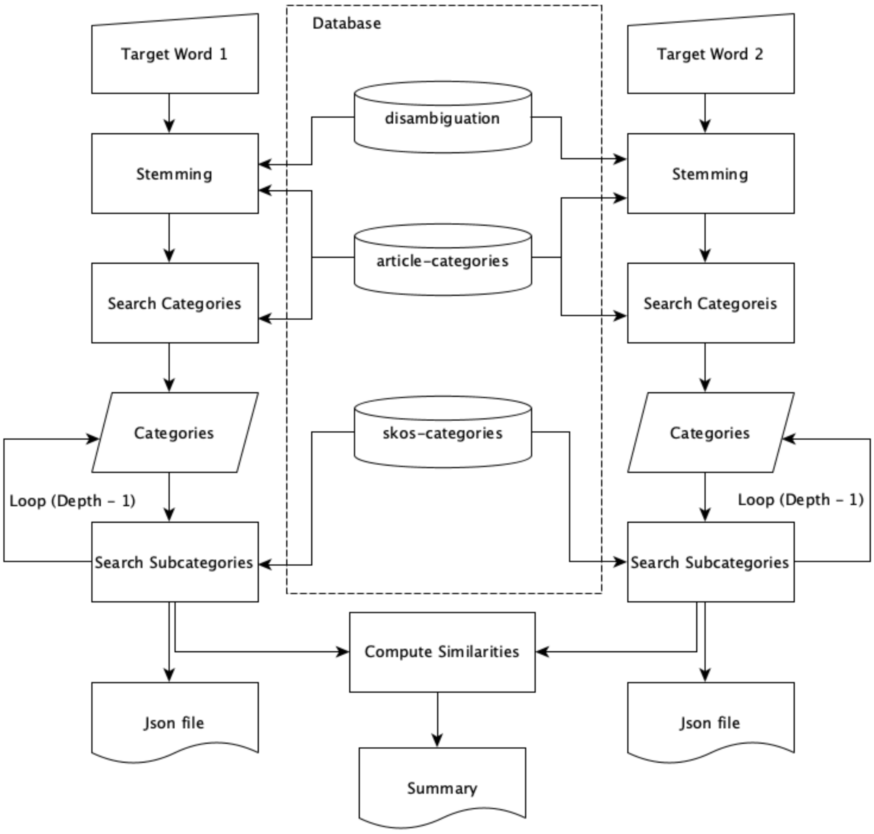


Figure 6: The flowchart of main program.

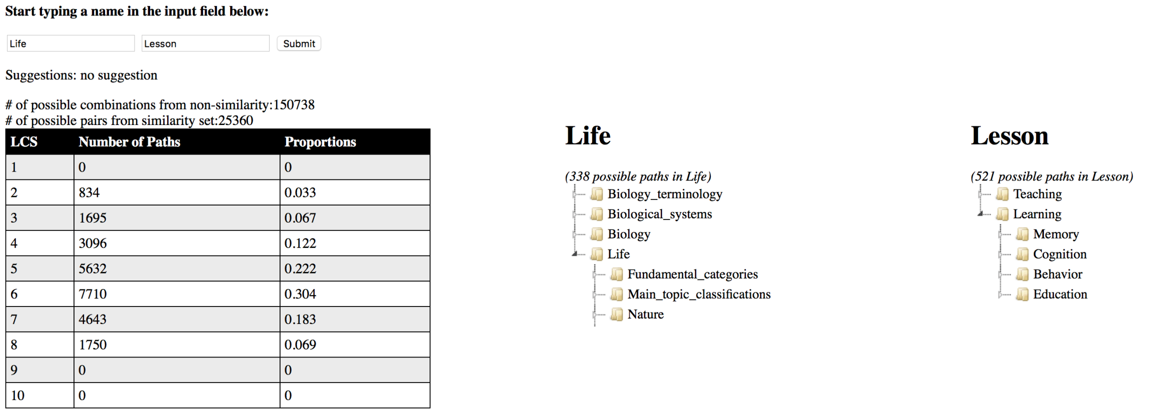
6 illustrated how we implement our method. After inputing a pair of target words (W_1, W_2), the program will start with stemming the words, and check if they can be found in *article-categories*. If not, the program will search the disambiguation database and return a category closest to the target word. After stemming, the program sends the linked categories as the input to Search Subcategories. This phase recursively searches superior categories of given categories until the search reaches the maximum number of depth we set initially. Once the search is done, the system generates a plain file in Jason format for displaying the output as a taxonomy-like graph on the website. Through the same procedure, the program generates the other plain file in the same format for the other target word. Finally, we use the distribution quantification method described earlier to generate the probability distribution $(p_0, p_1, \ldots, p_d, \ldots, p_\delta)$ for (W_1, W_2).

We developed a website to implement our method, http://ws.stat.sinica.edu.tw/wikiCat. Given a pair of words, it provides a summary table and two taxonomy-like graphs for the input words as shown in Figure 7. Every node in the graph represents a category, and it can be clicked to show its superior categories hidden underneath. The column "Proportions" gives the similarity distribution for the query (Life, Lesson). Compared to Figure 1, the agreement with the human raters is quite good. The time for executing a query varies around 2 seconds to 30 seconds.

4 Experiment

We use WordSim-353 to evaluate the performance of our method. We set $L = 5$ in order to be consistent with the scale used in WordSim-353 (from 0 to 10), so that our program will yield a probability distribution $(p_0, p_1, ..., p_{10})$ for each word-pair(W_1, W_2). To see how our probability distribution agrees with the score distribution of WordSim-353 raters, Kolmogorov-Smirnov statistic (K-S statistic) between two distributions is used. We perform the following procedure 1000 times to get a p-value. A p-value smaller than 0.05 indicates significant disagreement between the two distributions.

1. Simulating 13 (16, respectively) scores from



Start typing a name in the input field below:

Life | Lesson | Submit

Suggestions: no suggestion

# of possible combinations from non-similarity:150738
# of possible pairs from similarity set:25360

LCS	Number of Paths	Proportions
1	0	0
2	834	0.033
3	1695	0.067
4	3096	0.122
5	5632	0.222
6	7710	0.304
7	4643	0.183
8	1750	0.069
9	0	0
10	0	0

Life

(338 possible paths in Life)
Biology_terminology
Biological_systems
Biology
Life
Fundamental_categories
Main_topic_classifications
Nature

Lesson

(521 possible paths in Lesson)
Teaching
Learning
Memory
Cognition
Behavior
Education

Figure 7: A screen shot of the developed website.

the distribution $(p_0, p_1, ..., p_{10})$ for the word pair (W_1, W_2) from WordSim-353.1 (from WordSim-353.2, respectively).

2. Computing Kolmogorov-Smirnov distance between $(p_0, p_1, ..., p_{10})$ and the distribution of simulated scores.

After 1000 simulations, the p-value for (W_1, W_2) is given by the proportion of times that the observed K-S statistic exceeds the simulated K-S distance. As it turns, around 50% of word-pairs showed agreement between human rating and our computer rating (Figure 8). Given that the raters of WordSim-353 were from a generation before the inception of Wikipedia, we consider this result supports the potential of our folksonomy-based approach in reflecting human judgment diversity. Figure 9 showed some cases that our folksonomy-based method agreed very well with human rating.

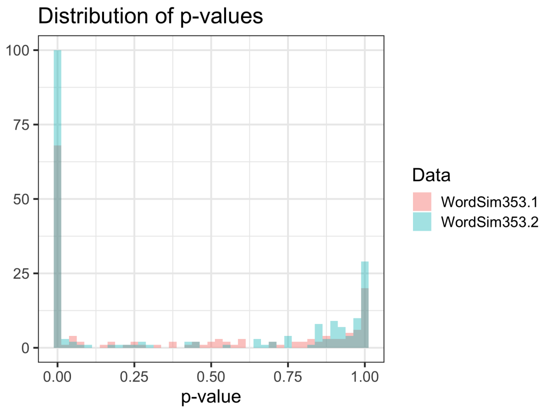


Figure 8: Histograms of p-values for WordSim-353.1 and WordSim-353.2. 53.59% of word-pairs have p-values greater than 0.05 in WordSim-353.1 and 48% in WordSim-353.2.

We further split the word pairs into two groups, AG (agreement, word pairs with p-value > 0.05) and DIS (disagreement, word pairs with p-value < 0.05). We examined the variance of human rater scores for each word-pair and plot the distribution for AG group and DIS group separately for comparison (Figure 10). We found AG group of word pairs tend to have larger variance than the DIS group. This indicates our approach may overestimate the degree of divergence in human rating, provided that the small group of raters participating WordSim-353 did not under-represent the true diversity of human behavior.

5 Application in Document Similarity Comparison

Our method can be extended for comparing documents. As a word can be mapped to multiple conceptual paths, a document will be mapped to an even bigger set of conceptual paths. As an example, we select three documents (*talk.politics.178908*, *talk.politics.178860* and *sci.med.59319*) from The 20 Newsgroups dataset (Lang, 1995). We further employed tf-idf (term frequency-inverse document frequency) (Salton and McGill, 1986) to extract the feature words of documents. Only top 10 words with highest tf-idf were kept (Table 2). We merge conceptual paths of these words to form a bigger set of representative conceptual paths for each document. Then we applied the same procedure as described in 3.2 to yield a probability distribution of similarity scores between two documents.

In this example, we set $L = 4$ to yield a probability distribution $(p_0, p_1, \ldots, p_8)$ for comparing two documents as shown in Table 3. Here PP is *talk.politics.178908* v.s. *talk.politics.178860*,

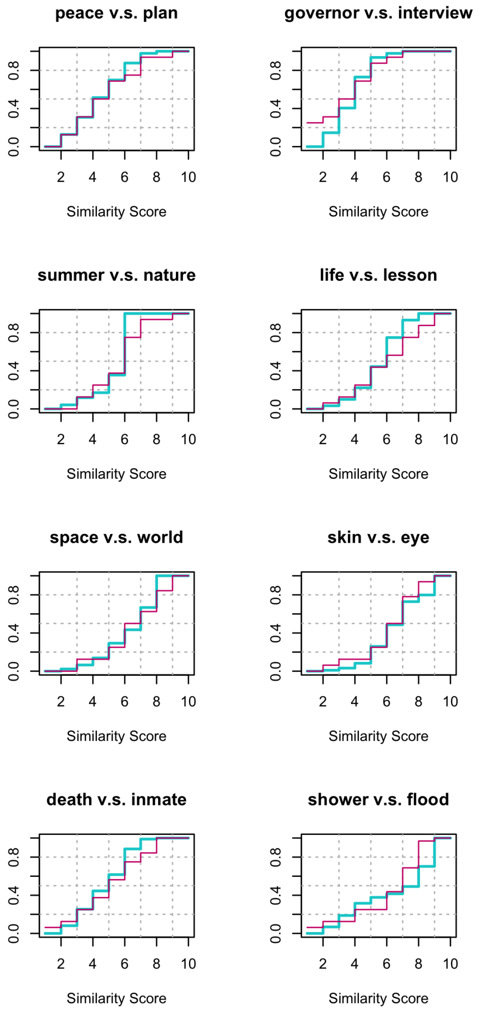


Figure 9: Eight cases that our method agreed well with human rating. The red lines are CDF by human rating and the blue lines are CDF by our folksonomy-based method.

talk.politics 178908	talk.politics 178860	sci.med 59319
president	oath	widex
masks	garrett	resound
attorney	gain	aids
federal	ingres	programmable
gas	nixon	hearing
reno	powers	loss
yesterday	office	ear
departments	personal	ahead
janet	monetary	sloping
children	indictment	reprogramed

Table 2: Lists of top 10 words with highest tf-idf scores.

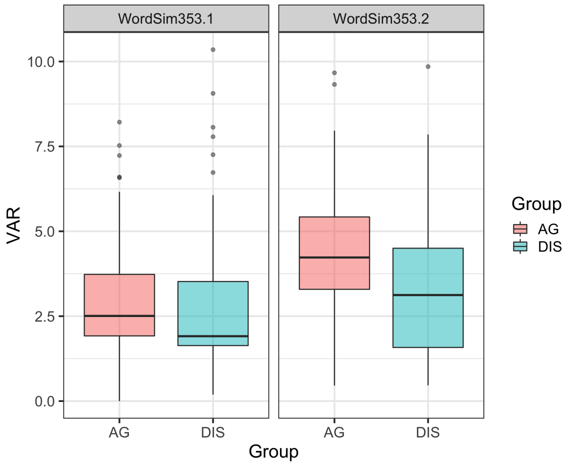


Figure 10: Boxplots for variances of similarity scores across 13 raters (WorSim-353.1) and 16 raters (WordSim-353.2). Word-pairs are split into two groups, AG (agreement, $p > 0.05$) and DIS (disagreement, $p < 0.05$).

	PP	PM1	PM2
0	0	0	0
1	0	0	0
2	0.1236742	0.2240363	0.2725498
3	0.1616162	0.3133787	0.3924248
4	0.1674242	0.245805	0.2225693
5	0.1511995	0.2126984	0.1124561
6	0.1440657	0.004081633	0
7	0.1337121	0	0
8	0.1183081	0	0

Table 3: Probability distributions of document similarity for comparing *talk.politics.178908*, *talk.politics.178860* and *sci.med.59319*.

PM1 is *talk.politics.178908* v.s. *sci.med.59319* and PM2 is *talk.politics.178860* v.s. *sci.med.59319*. Evidently, the probability distributions for (*talk.politics.178908*, *sci.med.59319*) and (*talk.politics.178860*, *sci.med.59319*) have low probabilities on high similarity scores (6, 7, 8). In contrast, we observe relatively higher probabilities being assigned to high similarity scores for (*talk.politics.178908*, *talk.politics.178860*).

6 Conclusion

Human perception on word similarity can be very discordant. Against the common trend of assigning a single score of similarity by most computer algorithms, we request a new computer task of assigning a probability distribution of similarity for each word pair. Leveraging the rich information embroidered behind the principle of free expression and empowered by user diversity of folksonomy, we design an approach that exploited the category tagging system of Wikipedia articles to perform the task. The good performance of our method is illustrated against two word similarity datasets with scores assigned by human raters. Our way of using Wikipedia (via folksonomy) is very different from many others; for example, the method of Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) treated articles in Wikipedia as a document corpus and produced only a single similarity score. For future works, we plan to modify our word similarity scoring formula by path-dependent weight adjustment for broadening the application in document comparison. It would also be worthwhile to apply our method to other languages for comparing the possible differences between languages in assigning similarity distributions.

Acknowledgments

This work was supported in part by grants from Academia Sinica, Taiwan, AS-104-TP-A07 and National Science Foundation, USA, NSF, DMS-1513622, and by MIB, Institute of Statistical Science, Academia Sinica. The content is solely the responsibility of the authors and does not necessarily represent the official views of NSF.

References

Amos Tversky. 1977. Features of Similarity. *Psycological Review*, 84(4):327–352.

Andrea Rodriguez and Max J Egenhofer. 2003. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456.

Aniket Kittur and Ed H. Chi, Bongwon Suh. 2009. What's in Wikipedia?: mapping topics and conflict using socially annotated category structure. In *The SIGCHI Conference on Human Factors in Computing Systems*, pages 1509–1512.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

Christopher D. Manning. 2015. Computational Linguistics and Deep Learning. *Computational Linguistics*, 41(4):701–707.

Euripides G.M. Petrakis, Giannis Varelas, Angelos Hliaoutakis, and Paraskevi Raftopoulou. 2006. X-Similarity: Computing Semantic Similarity between concepts from different ontologies. *Journal of Digital Information Management*, 4(4):233–237.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

Firt, J. R. 1957. A Synopsis of Linguistic Theory 1930-55. *Studies in Linguistic Analysis(special volume of the Philological Society)*, pages 1–32.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mohamed Ali Hadj Taieb, Mohamed Ben Aouicha, and Abdelmajid Ben Hamadou. 2014. Ontology-based Approach for Measuring Semantic Similarity. *Eng. Appl. Artif. Intell.*, 36(C):238–261.

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. 2015. Semantic Similarity from Natural Language and Ontology Analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Nilupulee Nathawitharana, Damminda Alahakoon, and Daswin De Silva. 2016. Using semantic relatedness measures with dynamic self-organizing maps for improved text clustering. *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2662–2671.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Nadella Sandhya and A. Govardhan. 2012. Analysis of Similarity Measures with WordNet Based Text Document Clustering. In *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012*, pages 703–714. Springer Berlin Heidelberg.

T. Slimani, B. Ben Yaghlane, and K. Mellouli. 2006. A New Similarity Measure based on Edge Counting. In *World Academy of Science, Engineering and Technology*, volume 17, pages 232–236.

Thomas K Landauer, Peter W. Foltz, and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Workshop at International Conference on Learning Representations*.

Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, and Xianyu Bao. 2015. A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications*, 42(4):2264–2275.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Yuhua Li, Zuhair A. Bandar, and David McLean. 2003. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):871–882.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *ACL 94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics Stroudsburg.