Fall 9-11-2020

Similarity Measures and Distance Measures Applications: A Software Engineering Prospective

Osama Rabie
King Abdulaziz University, obrabie@kau.edu.sa

Ehab Abozinadah
King Abdulaziz University, eabozinadah@kau.edu.sa

RESEARCH-IN-PROGRESS: SIMILARITY MEASURES AND DISTANCE MEASURES APPLICATIONS: A SOFTWARE ENGINEERING PROSPECTIVE

Osama Rabie
King Abdulaziz University
obrabie@kau.edu.sa

Ehab Abozinadah
King Abdulaziz University
eabozinadah@kau.edu.sa

ABSTRACT

In this paper, several applications that use similarity and/or distance measures. The review is mainly focusing on the prospective of systems design and analysis (i.e. software requirements analysis, interface design, software maintenance, etc.).

KEYWORDS

Similarity Measures, Distance Measures, Software Engineering Prospective

INTRODUCTION

A brief discussion on some of the challenges facing those applications and application evaluation of each application are included as well. A basic explanation of main concepts is given (see section 1.1). The applications reviewed are classified into intelligent application, software engineering enhancing application, and hybrid application. The applications are WordNet::Similarity (Pedersen et al. 2004), requirements-analysis supporting system (Park et al. 2000), and ROSA (Girardi et al. 1994) respectively. The types, equations, and in depth discussion of the similarity measures is beyond the scope of this paper.

Binary Similarity Measures and Binary Distance Measures

Binary Similarity Measures and Binary Distance Measures are measures used to determine the level of similarity between two (i.e. binary) patterns, or the level of dissimilarity (distance) between the patterns (Choi et al. 2010). There are many applications for these measures (e.g. clustering). Many fields are using the measures to help them deciding on the level of sameness between two patterns. Fields like ecology (Jackson et al. 1989), biometrics (Willett 2003), criminal justice (handwriting recognition) (Cha et al. 2002; Cha et al. 2003), etc.

INTELLIGENT APPLICATION

The class of intelligent applications contains an application that uses similarity measures in to be applied into more (pure) artificial intelligent filed. However, it is not to say that the use of similarity measures in any application does not have an artificial intelligent flavor to it. Again, the application uses the output of similarity measures to use it in order to produce artificially intelligent results. For the purpose of this paper, one application is discussed in this class: WordNet::Similarity (Pedersen et al. 2004).

WordNet::Similarity (Pedersen et al. 2004)

WordNet is a lexical database that is meant to produce the concept of a word (or words) and its interlink with other words semantically. WordNet::Similarity is a software package that can be downloaded free from (http://search.cpan.org/dist/WordNet-Similarity) or by SourceForge (http://wn-similarity.sourceforge.net). WordNet::Similarity is based on the WordNet however; it uses similarity measures to produce results that are more efficient than results from the WordNet.

The WordNet::Similarity uses modules written in Perl. WordNet::Similarity's Perl modules can be used by Perl programs (Pedersen et al. 2004). WordNet::Similarity uses six similarity measures and three measures of relatedness (Pedersen et al. 2004). WordNet is more compatible with similarity measures since the WordNet is based on the is-a (or hierarchy) relationship (Pedersen et al. 2004). However, the is-a (or hierarchy) relationship does not "cross part of speech boundaries" (e.g. no similarity to be measured between a noun and a verb). Therefore, measures of relatedness are used in WordNet::Similarity since the measures of relatedness include about any relationship between

two concepts including the is-a relationship. Discussing the functionality and usage of WordNet and measures of relatedness are beyond the scope of this paper.

Now, let us discuss the software package of WordNet::Similarity from the software engineering prospective. The package takes two concepts as the input and gives a number (i.e. the degree of similarity or relatedness) as the output. The package consists of Modules in Perl, which increase the user resistance that already can exist for using new systems (Aladwani 2001). In other words, from the prospective of system implementation not enough marketing (Aladwani 2001) is done to help in the system being adapted.

There are three ways to use the modules of the WordNet::Similarity. The modules can be used via the command line interface or web interface provided by a Perl program that can be found in the utils folder, which called similarity.pl. In addition, the Perl Modules can be embedded into a Perl program. The interface of the WordNet::Similarity is a command line interface for Perl that makes it inconvenient to be used by common people. Furthermore, one should build the program in Perl to be able to use it. The web interface can be dull and not very responsive. Not to mention that it can be a challenge to use the provided interface since it requires pre knowledge of the syntax. As a result, it is recommended to use the WordNet::Similarity software package through an application that uses a better interface and can handle Perl. However, arguably, by using Perl modules the first principle of the general software maintenance ethics that states "least advantage. Don't increase harm to the least advantaged" was violated (Collins et al. 1994). It was violated because not many people will be able to use Perl and (probably) fewer will be able to use the interface provided in the WordNet::Similarity (without spending the time going through the documentation). Furthermore, from user interface design prospective, they did not well serve the different needs of users and that made the WordNet::Similarity less likely to be adapted by users (Vasilyeva et al. 2005).

In addition, according to the distributers' website (http://search.cpan.org/dist/WordNet-Similarity/) the WordNet::Similarity software package was not updated or maintained since 16 Jun 2008. Although some researcher seemed to be interested in the package (Baldwin et al. 2003; Diab 2003; Jarmasz et al. 2003; McCarthy et al. 2004; Pedersen et al. 2004; Zhang et al. 2003), no literature newer than 2004 on the package can be found (at least at Microsoft Academic Search and Google Scholar). This is can be considered unpleasant from the prospective of systems maintenance. However, there are three assumptions on the reason for the WordNet::Similarity software package to be no-more. First, maybe because of the bad interface as discussed. Second, maybe it is because of the limitations of WordNet itself (Bond et al. 2012; Boyd-Graber et al. 2006; Budanitsky et al. 2006; Clark et al. 2006; Miller 1995; Navigli 2006; Oltramari et al. 2002; Prakash et al. 2007). Finally, the most likely reason to be the real reason behind the lake of interest in the WordNet::Similarity software package is the introduction of the semantic web in 2001 (Berners-Lee et al. 2001) and the start of the semantic web implementation (roughly) in 2006 (Shadbolt et al. 2006). This maybe caused researchers interested in the field to move towards another technology (i.e. semantic web) and leave the WordNet. Anyway, the WordNet::Similarity software package is no longer maintained and it looks like there will not be newer versions of the WordNet nor the WordNet::Similarity software package.

In conclusion, the WordNet::Similarity software package is more efficient and capable of being used to evaluate more diverse concepts than the WordNet. On the other hand, the package has some challenges that are needed to be addressed in order for the package to be used. However, it is worth mentioning that one of the main challenges is the limitation of the WordNet technology itself, which (assumingly) made researchers interested in the field became more interested in other technologies (e.g. the semantic web) instead.

SOFTWARE ENGINEERING ENHANCING APPLICATION

The class of software engineering enhancing application discusses an application (requirements-analysis supporting system (Park et al. 2000) that is using similarity measure(s) to improve the job of the software engineer and (theoretically) make it easier and more efficient.

Requirements-Analysis Supporting System (Park et al. 2000)

The requirements-analysis supporting system is a system that is meant to use similarity measures to between the sentences of system requirements. Ideally, the output should contains two aspects. First, "redundancies and inconsistencies" (Park et al. 2000). Second, discover hidden requirements, especially from ambiguous sentences (Park et al. 2000). The requirements-analysis supporting system consists of three modules. First, a module to measure the similarity between documents that consists of two sub-modules (Park et al. 2000). A sub-module to use the sliding window and another to use a parser that uses the syntactic dependency relations (Park et al. 2000). The intention behind the first module is to index a document and produce the information related to co-occurrence (Park et al. 2000). Second module is to measure the similarity between sentences (Park et al. 2000). This module is to check for

consistency between high-level and low-level requirements (Park et al. 2000). Finally, a module to identify ambiguous sentences (Park et al. 2000). This module uses Part-Of-Speech (POS) tagger to find and extract ambiguous sentences from the requirements (Park et al. 2000). Discussing the functionality and usage of sliding window method, parser, syntactic dependency relations, co-occurrence, high-level and low-level requirements, and Part-Of-Speech (POS) tagger are beyond the scope of this paper.

Now, let us discuss the software package of requirements-analysis supporting system from the software engineering prospective. The requirements-analysis supporting system is in client-server architecture (Park et al. 2000). The authors of the requirements-analysis supporting system's paper stated that they have used C as the programming language of the server-side and Visual Basic for the client-side to test the system (Park et al. 2000). As mentioned, the system consists of three modules. The first two (i.e. a module to measure the similarity between documents and a module to measure the similarity between sentences) take two inputs documents and sentences respectively. The first module output is the measurement (or degree) of similarity between the two documents (Park et al. 2000). The second module output is the measurement (or degree) of similarity between the two sentences (Park et al. 2000). Finally, the last module takes only one input (document) and gives only one output (list of ambiguous sentences) (Park et al. 2000).

Keeping in mind that the requirements-analysis supporting system wasn't found to be tested for the purpose of this paper, the literature claims it has high recall and precision (Park et al. 2000). This is can be claimed as a good match of the system requirements. However, from system implementation standpoint the requirements-analysis supporting system maybe did not do that good job for two reasons. First, the fact that it cannot be found makes a bad marketing for the system and that increases the users resistance (Aladwani 2001). Second, it is indicated from a newer literature (Ko et al. 2007) that the system creators are more interested in the theory than making/creating a software. Nonetheless, it is suggested that WordNet::Similarity (Pedersen et al. 2004) done a better job when it comes to these two points. WordNet::Similarity is easy to find and its authors included a link for it, and WordNet::Similarity has more literature. All that means the WordNet::Similarity has done better marketing (Aladwani 2001) than the requirements-analysis supporting system. The two points can be viewed negatively from the prospective of system maintenance.

By looking at the snapshots included in the literature, it looks like the requirements-analysis supporting system uses GUI. However, it does not seem to have the drag-and-drop feature.

One of the challenges in trying to implement the requirements-analysis supporting system that it uses a Korean parser (for the original system (Park et al. 2000)). This made the interface harder to understand and evaluate. However, it is unclear to state the usage of the Korean parser as a violation of the first principle of the general software maintenance ethics that states "least advantage. Don't increase harm to the least advantaged" (Collins et al. 1994).

In conclusion, according to the literature available, requirements-analysis supporting system is claimed to be sufficient. However, the aspects of system maintenance and user resistance ought to be better addressed.

HYBRID APPLICATION

The class of hybrid application contains an application that can be classified as intelligent application and software engineering enhancing application (in other words hybrid). In this class, the application reviewed called ROSA (Girardi et al. 1994).

ROSA (Girardi et al. 1994)

ROSA is mainly consists of two mechanisms classification and retrieval (Girardi et al. 1994). The classification mechanism takes two inputs (i.e. a software description in natural language and the software components) (Girardi et al. 1994). After that "The system extracts lexical, syntactic and semantic information and this knowledge is used to create a frame-like internal representation for the software component" (Girardi et al. 1994). The retrieval mechanism takes query as input and searches the knowledge base produced by the classification mechanism. Again, going through the technical details that are unrelated to software engineering is beyond the scope of this paper.

Now, let us discuss the software package of ROSA software from the software engineering prospective. It is built by using C as the programming language. By looking at literature and comparing ROSA with similar applications at its time, it can be indicated that ROSA did a better job meeting the efficiency requirements. However, there is no trace for ROSA (although some unrelated applications in the field of software engineering are called ROSA, too). In addition, by searching through Microsoft Academic Search and Google Scholar, no further literature can be found.

Apparently, ROSA did worse than WordNet::Similarity (Pedersen et al. 2004) and requirements-analysis supporting system (Park et al. 2000) when it comes to marketing and overcoming users resistance (Aladwani 2001).

Giving the time ROSA was used via a command line interface seems to be acceptable. Moreover, it does not seem to violate any of the general software maintenance ethics (Collins et al. 1994). By the end of the day, one should use whatever he can based on his best knowledge at the time of making the decision. In addition, the fact that ROSA was first built as a prototype in BIM- prolog (Girardi et al. 1994) gives a point for ROSA from the prospective of system Testing and Evaluation. That is ROSA was prototyped to test aspects separately and that is highly recommended (Mills et al. 1988).

In conclusion, ROSA can be considered as an efficient application in its field and time. On the other hand, the challenges from the prospective of system maintenance and marketing ought to be addressed.

FINAL REMARKS

In this paper, three applications that use similarity measures were reviewed, discussed, and critiqued from the prospective of software engineering. The applications discussed are WordNet::Similarity (Pedersen et al. 2004), requirements-analysis supporting system (Park et al. 2000), and ROSA (Girardi et al. 1994). The applications were classified into intelligent application, software engineering enhancing application, and hybrid application to give a better understanding of each application.

It is good to mention that applications using similarity measures almost do not exist in literature. More publications ought to be made to address this issue. More documentations about applications using the measures should be delivered as well. The hardest part of preparing this paper is the effort spent in finding applications (software wise) that uses similarity measures and have literature on these applications.

Finally, this paper contains the only applications (software wise) that use similarity measures and which their literature can be found at Microsoft Academic Search and Google Scholar. It is good to admit that and due to the availability challenge the review of WordNet::Similarity was better than the review of the rest two (i.e. requirements-analysis supporting system and ROSA). Apparently, it is hard to evaluate based on what others do think and without personally trying and evaluating. It is kind of like the Arabic saying “that who saw is not like who heard.”

REFERENCES

1. Aladwani, A. M. 2001. "Change management strategies for successful ERP implementation," *Business Process management journal* (7:3), pp 266-275.
2. Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. Year. "An empirical model of multiword expression decomposability," Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18, Association for Computational Linguistics2003, pp. 89-96.
3. Berners-Lee, T., Hendler, J., and Lassila, O. 2001. "The semantic web," *Scientific american* (284:5), pp 28-37.
4. Bond, F., and Paik, K. 2012. "A Survey of WordNets and their Licenses," *Small* (8:4), p 5.
5. Boyd-Graber, J., Fellbaum, C., Osherson, D., and Schapire, R. Year. "Adding dense, weighted, connections to WordNet," Proceedings of the Third International WordNet Conference2006, pp. 29-36.
6. Budanitsky, A., and Hirst, G. 2006. "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics* (32:1), pp 13-47.
7. Cha, S.-H., and Srihari, S. N. 2002. "A fast nearest neighbor search algorithm by filtration," *Pattern Recognition* (35:2), pp 515-525.
8. Cha, S.-H., Tappert, C. C., and Srihari, S. N. Year. "Optimizing binary feature vector similarity measure using genetic algorithm and handwritten character recognition," Proceedings of the Seventh International Conference on Document Analysis and Recognition, Citeseer2003, p. 662.
9. Choi, S.-S., Cha, S.-H., and Tappert, C. 2010. "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics* (8:1), pp 43-48.
10. Clark, P., Harrison, P., Jenkins, T., Thompson, J., and Wojcik, R. Year. "From WordNet to a Knowlege Base," Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering. Papers from the 2006 AAAI Spring Symposium2006, pp. 10-15.

11. Collins, W. R., Miller, K. W., Spielman, B. J., and Wherry, P. 1994. "How good is good enough?: an ethical analysis of software construction and use," *Communications of the ACM* (37:1), pp 81-91.
12. Diab, M. T. 2003. "Word sense disambiguation within a multilingual framework,").
13. Girardi, M., and Ibrahim, B. 1994. "A similarity measure for retrieving software artifacts," *University of Geneva, Centre Universitaire d'Informatique*).
14. Jackson, D. A., Somers, K. M., and Harvey, H. H. 1989. "Similarity coefficients: measures of co-occurrence and association or simply measures of occurrence?," *American Naturalist*), pp 436-453.
15. Jarmasz, M., and Szpakowicz, S. Year. "S.: Roget's thesaurus and semantic similarity," In: Proceedings of the RANLP-2003, Citeseer2003.
16. Ko, Y., Park, S., Seo, J., and Choi, S. 2007. "Using classification techniques for informal requirements in the requirements analysis-supporting system," *Information and Software Technology* (49:11), pp 1128-1140.
17. McCarthy, D., Koeling, R., and Weeds, J. 2004. "Ranking WordNet senses automatically," *recall* (40), p 60.
18. Miller, G. A. 1995. "WordNet: a lexical database for English," *Communications of the ACM* (38:11), pp 39-41.
19. Mills, R., Meyer, E., Cathcart, G., and Clemens, L. Year. "A user-assisted test and evaluation methodology assistant program (TEMAP)," Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National, IEEE1988, pp. 1060-1064.
20. Navigli, R. Year. "Meaningful clustering of senses helps boost word sense disambiguation performance," Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics2006, pp. 105-112.
21. Oltramari, A., Gangemi, A., Guarino, N., and Masolo, C. 2002. "Restructuring WordNet's top-level: The OntoClean approach," *LREC2002, Las Palmas, Spain* (68).
22. Park, S., Kim, H., Ko, Y., and Seo, J. 2000. "Implementation of an efficient requirements-analysis supporting system using similarity measure techniques," *Information and Software Technology* (42:6), pp 429-438.
23. Pedersen, T., Patwardhan, S., and Michelizzi, J. Year. "WordNet:: Similarity: measuring the relatedness of concepts," Demonstration Papers at HLT-NAACL 2004, Association for Computational Linguistics2004, pp. 38-41.
24. Prakash, R. S. S., Jurafsky, D., and Ng, A. Y. 2007. "Learning to Merge Word Senses," *Computer Science Department Stanford University*).
25. Shadbolt, N., Hall, W., and Berners-Lee, T. 2006. "The semantic web revisited," *Intelligent Systems, IEEE* (21:3), pp 96-101.
26. Vasilyeva, E., Pechenizkiy, M., and Puuronen, S. Year. "Towards the framework of adaptive user interfaces for eHealth," Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on, IEEE2005, pp. 139-144.
27. Willett, P. 2003. "Similarity-based approaches to virtual screening," *Biochemical Society Transactions* (31), pp 603-606.
28. Zhang, Z., Otterbacher, J., and Radev, D. Year. "Learning cross-document structural relationships using boosting," Proceedings of the twelfth international conference on Information and knowledge management, ACM2003, pp. 124-130.