IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 23-26

A novel approch Automatically Categorizing Software Technologies.

Mr.Dinesh Bhere[1], Prof.Kunal Kore[2]
[1](Computer Department, Sharadchandra Pawar College of Engineering, India)
[2](Computer Department, Sharadchandra Pawar College of Engineering, India)

Abstract: *Software development is increasingly based on reusable components in the form of frames and libraries, as well as programming languages and tools to use them. Informal language and the absence of a standard taxonomy for software technologies make it difficult to reliably analyze technological trends in discussion forums and other online sites. The system proposes an automatic approach called Witt for the categorizationof software technology. Witt takes as input a sentence that describes a technology or a software concept and returns a general category that describes it (for example, an integrated development environment), along with attributes that qualify it even more. By extension, the approach allows the dynamic creation of lists of all technologies of a given type.*

I. Introduction

Now days the Software development is increasingly basedon reusable components platform in the form of frames and libraries, as well as programming languages and tools to use them. Taken together, these software technologies form a massive and rapidly growing catalog of constituent elements for systems that becomes difficult to monitor through discussion channels. The list of all technologies of a certain type or their popularity in relation to this type. Questions like ”what is the most popular web application framework?” They are important for many organizations, for example, to decide which development tool to adopt at the beginning of a project or for which technology to develop a driver. The answers to these questions are routinely proposed without any supporting data, but it is difficult to find valid empirical surveys. To move to a rationalized, evidence-based approach to monitor the use of software technologies, we must be able to automatically classify and group the nominated mentions of software technologies.

II. Material And Methods

In the proposed system, our approach takes as input a termto categorize. As a vocabulary for the software technologysystem, they have data of all the methodologies, so the systemgets the data labels. According to the label, they will obtainall the data coming from a different technology. Apply NLPand Levenshtein distance algorithm. Then hypernyms will findlike final step of the proposed system contains of transformingthe hypernyms into a set of categories, possibly with someattributes. this system designed categories to represent generalhypernyms, with a focus on coverage: commercial ide for phpis a better (more precise) hypernyms than ide, but the latter is a better category (higher coverage). The attributes are meantto provide a flexible way to express the information lost whentransforming a hypernyms into a category. They representtypical variants of the category, but would not constitute validhypernyms on their own. To transform a hypernyms intocategory with attributes, this system start by removing all noninformativephrases like name of and type of this system alsotransform phrases indicating a collection, e.g., set of, into theattribute collection of, and remove it from the hypernyms. Thissystem constructed a small list of such phrases based on ourdevelopment set. If two or more occurrences of the word of or
of the word for remain in the hypernyms, this system do notparse the hypernyms, as its structure is possibly too complexfor our simple heuristics.

A. Algorithms

1) NLP: Natural language processing (NLP) is a subfield ofcomputer science, information engineering and artificial
intelligence that deals with the interactions between computers and (natural) human languages, especially how to program computers to process and analyze large quantities of data. in natural language.
2) Similarity score: Calculate the string similar based on the similarity of the grams Q between the first paragraph of the section of the selected article and the extract of the label. The similarity is calculated for the first line of both texts, then the first two sentences, the first three, etc., till one of the inputs parameter runs out of

sentences. The best similarity score is keep representative of the overall similarity between the two inputs parameter.

3) Levenshtein distance algorithm: The Levenshtein distance is a string metric to measure the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. The Levenshtein algorithm:calculates the least numberof edit operations that are necessary to modify onestring to obtain another string. The most common way of calculating this is by the dynamic programming approach. In proposed system Present system using this to match user entered question with available question in database.

Input. : Get user entered question.

Working:

Step1. Select user entered query

Step 2:. Select all data from available database

Step3. Pass the distance to match query question with available data. System will check question with according to entered query with available data. word by word with available answer.

Step4:One by one query will gets by visiting each data to specified distance.

Output: Get matched similar data.

B. Mathematical Model : This will be used to calculate accuracy in proposed system.It categorize query and result data from system. In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query: precision=relevantdocument\retrieveddocuments retrieveddocuments

In information retrieval, recall is the fraction of the relevant documents that are successfully retrieved.

recall=relevantdocument\retrieveddocuments relevantdocuments

In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Statistical analysis

Data was analyzed using SPSS version 20 (SPSS Inc., Chicago, IL). Student's t-test was used to ascertain the significance of differences between mean values of two continuous variables and confirmed by nonparametric Mann-Whitney test. In addition, paired t-test was used to determine the difference between baseline and 2 years after regarding biochemistry parameters, and this was confirmed by the Wilcoxon test which wasa nonparametric test that compares two paired groups. Chi-square and Fisher exact tests were performed to test for differences in proportions of categorical variables between two or more groups. The level $P < 0.05$ was considered as the cutoff value or significance.

III. Result

Categorize, it gives efficient time to categorize document according to entered string.Fig.2-Graph showed a pictorial representation of No.of matched document time. X-Axis contains no.of document and y-axis time to match query.Graph shows in proposed system how search time varies with respect to the number of documents. in our implementation, search time depends not only on the number of documents returned, but also on the number of documents in which the query to be categorize are present.

Table I

Execution Time For Categorize The Entered Query In No.Of Documents.

Index	No of Documents	Query	Time to categorize(ms)
1	1000	Query 1	101
2	2000	Query 2	140
3	3000	Query 3	190

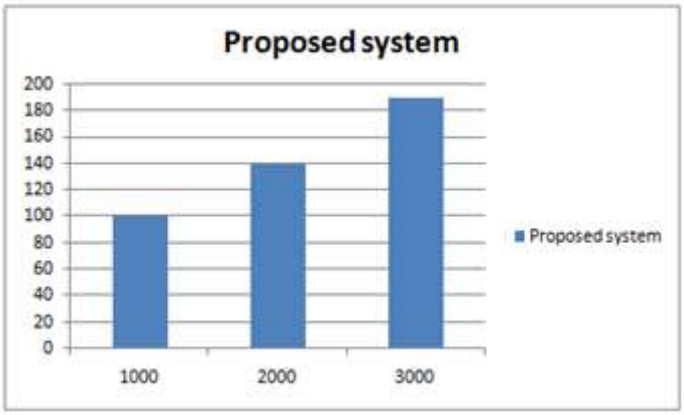

Fig. 2. Categorize time for no.of documents.

IV. Discussion

An important step towards understanding the terminology of the machine is the discovery of hypernyms, that is, the discovery of the more general concept in an is-a relationship (for example, AngularJS is a web application framework), which has led to the development of many tools hypernyms automated extraction. Unfortunately, the discovery of correct hypernyms is not efficient to support the detection and monitoring of comparable software techniques. For example, the cross-platform commercial IDE for PHP is a hyper valid for Php Storm, but the expression is too specific to make a useful category of technologies. The categorization of software technologies is a much more complex problem that requires greater abstraction and normalization.

In Software development contains more data frameworks and libraries, and the programming languages and tools to use them. Considered together, these software technologies form a massive and rapidly-growing catalog of building blocks for systems that becomes difficult to monitor across discussion channels.

System working in software categorization in future it can useful in medical system.

1)Important step toward the machine understanding of terminology is hypernym discovery, i.e., the discovery of the more general concept in a is-a relationship (e.g., AngularJS is a web application framework), which led to the development of many automated hypernym extraction tools. Unfortunately, discovering valid hypernyms is not sufficient to support the detection and monitoring of comparable software technologies. For example, commercial cross-platform IDE for PHP is a valid hypernym for PhpStorm, but the expression is too specific to constitute a useful category of technologies. Categorizing software technologies is a much more complex problem that requires additional abstraction and normalization.. Softwares are designed to be used a significant amount of time, therefore maintenance represents an important part of their life cycle. It has been estimated that a lot ofthe time allocated to software maintenance is spent on the program comprehension. Many approaches using the program structure or external documentation have been created to ease the program comprehension. However, another important source of information is still not widely used for this purpose: the identifiers. In this article, Present system propose an approach, based on Natural Language Processing techniques, that automatically extracts and organizes concepts from software identifiers in a WordNet-like structure: lexical views. Those lexical views give useful in- sight on an overall software architecture and can be used to improve results of many software engineering tasks. The proposal is validated on a corpus of 24 open source software's.

2. Measuring the similarity of words is important in accurately representing and comparing documents, and thus improves the results of many natural language processing (NLP) tasks. The NLP community has proposed various measurements based on WordNet, a lexical database that contains relation- ships between many pairs of words. Recently, a number of techniques have been proposed to address software engineering issues such as code search and fault localization that require understanding natural language documents, and a measure of word similarity could improve their results. However, WordNet only contains information about words senses in general-purpose conversation, which often differ from word senses in a software-engineering context, and the software-specific word similarity resources that have been developed rely on data sources containing only a limited range of words and word uses. In recent work, Present system have proposed a word similarity resource based on information collected automatically from StackOverflow. Present system have found that the results of this resource are given scores on a

3-point Likert scale that are over 50per higher than the results of a resource based on WordNet. In this demo paper, Present system review our data collection methodology and propose a Java API to make the resulting word similarity resource useful in practice.

V. Conclusion

System proposed a novel a domainspecifictechnique to automatically produce an attributed category structure describing an input phrase assumed to be a software technology. Here found that after transforming hypernyms into more abstract categories. Our approach takes as input a term to categorize software. It uses NLP and Levenshtein distance algorithm.

References

[1]. J.-R. Falleri, M. Huchard, M. Lafourcade, and M. Dao, Automatic extraction of a WordNet-like identifier network from software, in 18thIEEE International Conference on Program Comprehension (ICPC), 2010, pp. 413.
[2]. SEWordSim: Software-specific word similarity database, in Companion Proceedings of the 36th International Conference on Software Engineering, 2014
[3]. C. Treude and M.-A. Storey,Work item tagging: Communicating concerns in collaborative software development, IEEE Transactions on Software Engineering, vol. 38, no. 1, 2012.
[4]. M. F. Porter, An algorithm for suffix stripping, Program, vol. 14, no. 3, 1980.
[5]. G. A. Miller, R. Beckwith, D. Gross, and K. J. Miller, Introduction to Wordnet: An on-line lexical database, International Journal of Lexicography, vol. 3, no. 4, 1990.
[6]. A. K. Saha, R. K. Saha, A discriminative model approach for suggesting tags automatically for Stack Overflow questions, in Proceedings of the 10th Working Conference on Mining Software Repositories, 2013, .
[7]. R. Snow, D. Jurafsky, and A. Y. Ng, Learning Syntactic Patterns for Automatic Hypernym Discovery, in Proceedings of the 18th Annual Conference on Neural Information Processing Systems, 2004.
[8]. T. Wang, H. Wang, G. Yin, X. Li, and P. Zou, Tag recommendation for open source software, Frontiers of Computer Science, vol. 8, no. 1, , 2014.
[9]. J. Yang and L. Tan, SWordNet: Inferring semantically related words from software context, Empirical Software Engineering, pp. 131, 2013.
[10]. T. Zesch, C. Mller, Extracting lexical semantic knowledge from wikipedia and wiktionary, in Proceedings of the Conference on Language Resources and Evaluation, electronic proceedings, 2008