Mining Hypernyms Semantic Relations from Stack Overflow

László Tóth
Department of Software Engineering
University of Szeged
premissa@inf.u-szeged.hu

Balázs Nagy
Department of Software Engineering
University of Szeged
bnagy@inf.u-szeged.hu

Tibor Gyimóthy
MTA-SZTE Research Group on Artificial Intelligence
Department of Software Engineering
University of Szeged, Hungary
gyimi@inf.u-szeged.hu

László Vidács
MTA-SZTE Research Group on Artificial Intelligence
Department of Software Engineering
University of Szeged, Hungary
lac@inf.u-szeged.hu

ABSTRACT

Communication between a software development team and business partners is often a challenging task due to the different context of terms used in the information exchange. The various contexts in which the concepts are defined or used create slightly different semantic fields that can evolve into information and communication silos. Due to the silo effect, the necessary information is often inadequately forwarded to developers resulting in poorly specified software requirements or misinterpreted user feedback. Communication difficulties can be reduced by introducing a mapping between the semantic fields of the parties involved in the communication based on the commonly used terminologies. Our research aims to obtain a suitable semantic database in the form of a semantic network built from the Stack Overflow corpus, which can be considered to encompass the common tacit knowledge of the software development community. Terminologies used in the business world can be assigned to our semantic network, so software developers do not miss features that are not specific to their world but relevant to their clients. We present an initial experiment of mining semantic network from Stack Overflow and provide insights of the newly captured relations compared to WordNet.

CCS CONCEPTS

• **Computing methodologies → Semantic networks**.

KEYWORDS

semantic network, communication silos, mining, lexico-syntactic pattern, NLP, Stack Overflow, WordNet

ACM Reference Format:
László Tóth, Balázs Nagy, Tibor Gyimóthy, and László Vidács. 2020. Mining Hypernyms Semantic Relations from Stack Overflow. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3387940.3392160

KG4SE 2020, May 26, 2020, Seoul, South Korea

ACM ISBN 978-1-4503-7963-2/20/05…$15.00
https://doi.org/10.1145/3387940.3392160

1 INTRODUCTION

The most frequent issues of software projects are the requirements comprehension and the establishment of a common understanding among the different domain experts. The principal difficulty is the usage of the same terminologies with different meanings and contexts, along with the corresponding tacit knowledge, which is usually not articulated during the elicitation process. This dilemma exists between the developers and the customers in almost every communication situation.

The problem exists not only between different organizations, but also internally in a team leading to silo effects that have a detrimental impact on productivity and efficiency. Communication silos are well-studied phenomena in various fields of science, from organizational psychology to engineering [8, 13, 15, 41]. One of the biggest challenges in practical software engineering is to combat the communication silos in projects with multiple different participants from different organizations and/or domains not only in the requirements engineering [14, 39] but in every communication between the customers and the development team [43].

For the reconciliation of the different meanings and to catch the corresponding tacit knowledge, a mapping of the different semantic fields is needed. The notions used in a particular domain often provide another or overplus meaning of the words denoting them. The semantic field – also called the lexical field – is a set of lexemes describing a conceptual domain along with the relationships with each other [20, 22]. The semantic field can be represented by directed graphs where the nodes are the terms related to the specific notion, and the edges represent the relationship among the notions. These constructions are called semantic networks [34]. Semantic networks can also be used as a psychological model of the notions of the world in the human mind [5, 25]. With the aid of semantic networks, terms from different semantic fields can be matched together either directly or via a proper upper ontology [23, 26]. This mapping process supports the recognition of the different properties of the meaning and the catching of the tacit knowledge among the participant of different domains.

This paper presents a method that extracts the principal notions used by software developers and creates the corresponding semantic networks based on one of the most common semantic relationships called hypernyms (*'is a'* relationship). The dataset used for extraction is the collection of Stack Overflow (SO) posts.

The SO community has been making a significant effort to maintain the quality and professionality of the site [10, 38]. Stack Overflow is, on one hand, a community portal for programmers and, on the other hand, a knowledge repository aimed to provide professional support for programmers in their everyday work. Consequently, the language and the conversations on SO reflect the semantic field common among software developers making the textual data of posts suitable for this research.

In this paper we introduce a mining method applied to Stack Overflow to obtain a semantic network that represents specialized domain knowledge for software engineering. For the demonstration, we process 1.3 million randomly selected sentences from SO, and use WordNet, a lexical-semantic database for general English. Given the differences between the two networks, we argue that an extended network from the two sources could enhance understanding in the presence of a tacit knowledge. Although the experiments are performed on a sample data of SO, this preliminary analysis also provides an insight into the geek language illustrated by examples. The contributions of our work are the following:

- We present an algorithm for extracting the semantic relations from a specific database containing conversations and discussions. Our method is highly insensitive to the occasionally occuring weak usage of English.
- We build a semantic network from SO posts containing hypernyms, the most common semantic relationship.
- We demonstrate the usability of the extracted network in reconciling the discrepancies among the different domains using WordNet.

2 BACKGROUND

One of the essential purposes of semantic networks is to highlight the difference of a specific term in different semantic fields. Let us consider the following example: the term *value* means a worth of something or amount of something according to WordNet, whereas in software engineering, *value* can also be a reference to an object or the result of a function or even a reserved word.

In this section we briefly present our approach to represent semantic networks and introduce two sources of semantic knowledge: the WordNet network and the Stack Overflow website.

2.1 Semantic Networks

The human brain is an efficient information processing system considering its flexibility and adaptability. The human memory, particularly the long term memory, can be modeled in an organized form called semantic network [3, 5, 16, 24]. Semantic networks are graph structures where nodes represent the concepts via their terms, and edges represent the links among those concepts. We apply the definitional network approach [34], which is also closely parallel with the semantic memory in the human brain [24, 25]. This structure is a model of the super-subordinate relations or, as called in linguistics, hyperonym-hyponym relations. This bond is also typical in object-oriented analysis and design, which is the generalization-specification relationship, also called inheritance.

The relationship mentioned can be formulated as the following, based on the definition from Gabor Melli [1]: Let $\mathcal{WP}$ be the set of word-phrases (*words* and *group of words*) in a particular natural language, and let $NP \subset \mathcal{W}$ be the set of noun-phrases of that language. Let C be a set of concepts, that is, the elements of the mind to which the mapping of objects in the world exists, whether these objects are real or fictional. Let $X, Y \in NP$ and let $P : \mathcal{WP} \to C$ be a given mapping from the set of word-phrases to the set of concepts.

- **Hyponymy relation** holds between X and Y, denoted as $Hyponymy(X, Y)$ iff $WP(X) \Rightarrow WP(Y)$, but $WP(Y) \nRightarrow WP(X)$.
- **Hyperonymy relation**, denoted as $Hyperonymy(X, Y)$ is the inverse relation of the hyponymy relation.

These connection types provide proper insight into the notions used in a particular domain. The set of concepts used in that domain, along with their relationships, forms the semantic or lexical field [20]. The semantic field is defined as the set of concepts or ideas, where the concepts are symbolized using words and word structures, also called terms. These are models of objects in the world, whether real or fictional. The relationships among these notions can be represented via the semantic triangle shown in Figure 1[30].

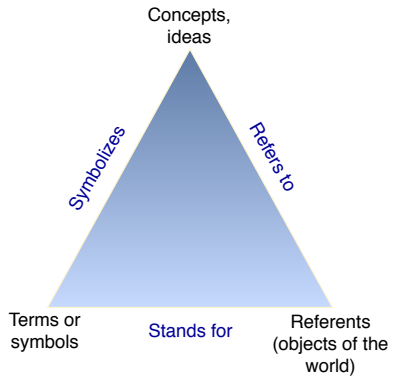

Figure 1: Semantic (Odgen/Richards) Triangle

Augmenting the above with the corresponding definition from [11], our particular semantic network can be formulated as follows:

- Let C be the set of concepts or the semantic domain. Let also be given the set of noun-phrases $NP \in \mathcal{WP}$ from a natural language and the mapping $P : \mathcal{WP} \to C$ from the set of word-phrases to the semantic domain.
- The semantic network is defined as the following directed graph of the terms: Let $G = (NP, E, s, d)$, where NP is the set of nodes (*the terms given as noun-phrases*), E is the set of edges. An edge $e \in E$ exists between $X, Y \in NP \Leftrightarrow (X, Y) \in \{Hyperonymy(X, Y)\}$. The $s, d : E \to \mathcal{P}(NP)$ are two functions, referred as the source and destination functions, and $\forall X, Y \in NP : (X, Y) \in E \Rightarrow (X \in s(e) \wedge Y \in d(e))$.

We consider only the hyperonym (*and the implicitly given hyponym*) relations among the noun-phrases in our investigation.

2.2 WordNet

WordNet is an extensive lexical-semantic database of English. The different words are grouped into sets of synonyms called synsets. In WordNet, the synsets build those symbols that are mapped to the set of concepts. The definition of the given concepts is also provided.

[1] http://www.gabormelli.com/RKB/Hyponymy_Relation

There are 117,000 synsets, and each of them is connected to other synsets based on various types of relations, like: Super-subordinate relations also called hyperonymy and hyponymy among the noun-phrases, meronymy (*and implicitly the holonymy*), which represents the part—whole relationships among the synsets and antonymy by which adjectives are organized. For complete view we refer to [28].

WordNet is a semantic network based on a directed hypergraph in which the nodes are synsets, and edges represent the relations mentioned above. The semantic network obtained in this present study can be linked to WordNet by implementing a proper mapping of the set of nouns in the noun-phrases (*NP*) vertices of our semantic network to the synsets of WordNet containing the particular noun. Using this mapping, we can discover an extended semantics of a particular term using the relationships from the corresponding synsets. WordNet, therefore, can play as an interface among the different semantic fields.

2.3 Stack Overflow

Stack Overflow is a Q&A site with the main purpose of providing answers to clear and well-described questions; SO is not intended to support long discussions and chats [4]. Posts that fail to satisfy the quality requirements established by the community are to be closed and eventually deleted [38]. For closing or deleting posts, privileged users can submit votes. Privileges are based on a reputation system, where reputation points can be obtained by user activities. The popularity of SO is well reflected by the 8,000 new questions every workday. Based on the data published by SO on January 3, 2020, the portal has more than 53 million monthly visitors, more than 18 million questions, and more than 28 million answers [2].

Stack Overflow can be considered as an extensive knowledge repository in the software development domain. The reviewed posts contain the knowledge and terminology used *de facto* among the developers. This feature allows the mining of the semantics of terms in this specific domain. Nevertheless, some critical issues should be considered. Most of the posts contain code fragments, links, and special characters. Besides, incorrect grammar also appears in some cases, and appropriate evaluation of these posts can be ambiguous for the community[3]. We apply special heuristics in this work to filter out or correct false relationships, but we note that the most accurate correction is undoubtedly the expert human review.

3 MINING PROCESS

The goal of the data mining process is the recognition and extraction of terms along with their hypernym relations from the noisy dataset of Stack Overflow. After the extraction we compose the semantic contexts of terms in the form of a semantic network. The overall workflow for the mining process can be seen in Figure 2. Posts used in the mining process are extracted from the Stack Overflow data dump created on March 4, 2019. We migrated this dump into a PostgreSQL 10.10 database and used it for further processing. The total number of imported posts is 43,872,992, and from these, the amount of questions is 17,278,709. In this experiment we used a sample of questions containing 1.3 M sentences randomly selected after the preprocessing phase.

[2]https://stackoverflow.com/company
[3]https://meta.stackoverflow.com/questions/253780

3.1 Preprocessing the raw text

The objective of the preprocessing is to provide a cleaned input text sliced into sentences for the mining process. The raw text contains various elements that we first have to eliminate or simplify. Besides, we have to ensure that the processed text contains only relevant and accepted questions and answers. Therefore we only consider posts obtained non-negative scores from the Stack Overflow community. The preprocessing consists of the following steps:

- Stack Overflow posts with non-negative scores are collected.
- Code blocks are replaced with the string `code example`.
- Hyperlinks are replaced with the string `link`.
- Text is cleaned from the HTML tags.
- `C++`, `C#`, and `F#` as well as their lower case counterparts are replaced with `cplusplus`, `csharp`, and `fsharp`, respectively.
- Some special characters are replaced with white space characters but punctuation is retained.
- Multiple white spaces are combined into a single space.
- The text is then split into sentences.

We applied the Python Regex and BeautifulSoup packages. The resulting sentences were written to a comma separated text file (CSV), where each row corresponds to a single processed sentence. 137,441,012 sentences were prepared for the extraction process.

3.2 Mining semantic relations

Our extraction procedure follows the method proposed by Hearst [17], who introduced lexico-syntactic patterns to catch hyponym relations from free texts. These patterns were later extended based on the patterns observed on various Web pages [33]. In our experiments, we apply the following two common patterns for extraction:

- $\{NP_t\}$ $(which)$ $\{is|was|are|were\}$ $\{NP_h\}$
- $\{NP_h\}$ $\{for\ example|e.g.|i.e.\}$ $\{NP_t\}$

NP_t denotes the hyponym part of the pattern, whereas NP_h means the hypernym member. We use parentheses for indicating optional components of the pattern, and pipe (|) is used for signifying a choice among more than one constituents.

For parsing a given input sentence, we applied the parser from the module `pattern.en` by the Computational Linguistics & Psycholinguistics Research Center [37]. The parser provides tokenization, part-of-speech (POS) tagging, and chunking, i.e., grouping consecutive words that belong together.

In the first stage of the transformation process, the words of the sentences are checked against the set of valid English words. This checking is due to the weak POS tagging of the technical terms, which are considered as a proper noun. The annotated sentences are then transformed into masked sentences, where noun-phrases are replaced by the string 'NP', and the corresponding tracking information is saved. This is necessary for writing back the noun-phrases later in the resulting relationships. As the input contains occasional sentences with incorrect grammar and wrong punctuation, the parser often splits the unified parts of noun-phrases into smaller pieces. Besides, some words from the original noun-phrases, such as the preposition *of*, were omitted from the phrases. Some possible word parts of noun-phrases have to be checked whether they are parts of lexico-syntactic patterns, and handled appropriately. In our case, the word 'example' must be checked whether it is part of the

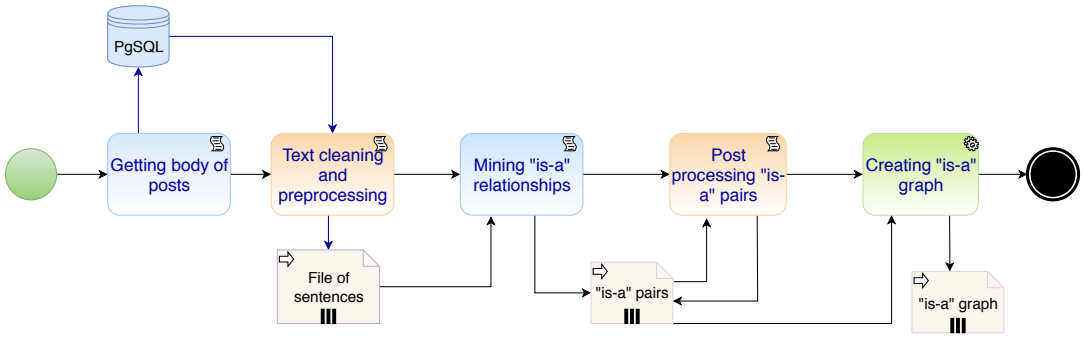

Figure 2: Mining process

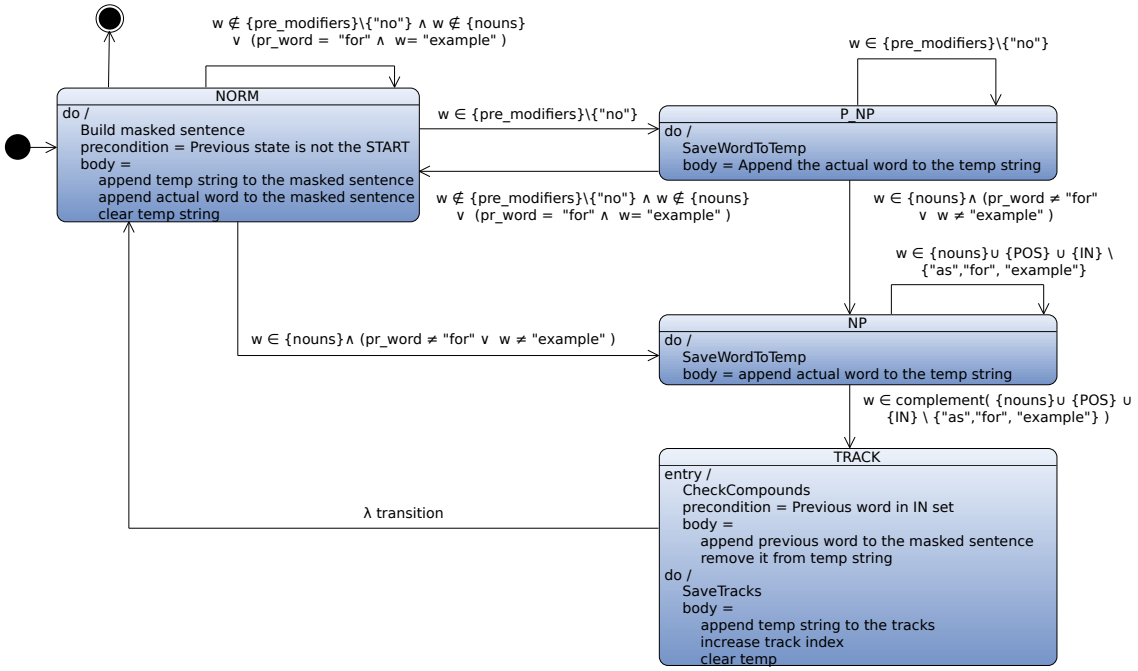

Figure 3: Extracting noun phrases

phrase 'for example' and whether it follows a noun or something else. In these cases, 'example' is not part of the noun-phrases of our interest. The procedure is presented in Figure 3.

After the above step, the input is ready for the extraction process, which is performed using regular expressions written on the basis of the lexico-syntactic patterns. Note that the extracted strings contain the string 'NP'. These placeholders have to be converted back to the corresponding original noun-phrases using the tracking information saved in the preprocessing phase.

During the extraction process, the structure of the second noun-phrases is examined. In case of the pattern $\{NP_t\}$ $\{is\}$ $\{NP_h\}$ the very first word of the NP_h has to be a determiner. Those results that do not follow this rule are filtered out.

The resulting noun-phrases are appended in a CSV file, where the first element of each line corresponds to the hyponym part followed by the hypernym part of the specific relation.

3.3 Postprocessing of the relations

As noted, the input may contain sentences using poor English grammar. Word order errors have disturbing effects on the extraction process because this kind of mistakes might reverse the noun phrases in the patterns, and the process results in mixed (*the intended and their inverse*) relations. We applied heuristics to mitigate this effect. We consider a proper noun as the hyponym part of a given relation if the alternative member of the pair is other than a proper noun. The remained pairs are checked against WordNet using the noun parts of the phrases for mapping. If WordNet suggests the inverse relations, the elements of the pairs are reversed.

Although the input was thoroughly preprocessed and cleaned, some unique strings still remained in the input text that have to be removed from the results. Texts related to software engineering often contain single letters denoting variables and constants. If the

extraction process recognizes them as part of a hyponym relationship, we have to remove them because they are not a general type of any of these relations. Similarly, the possessive personal pronouns specify the relations to a particular case, therefore they have to be removed as well. The cleaned pairs were kept only in the case of a real relationship; the left and right sides have to contain different nouns along with their modifiers.

4 RESULTS AND DISCUSSION

The resulting pairs of the extraction are the edge list of the semantic network under investigation. This network is represented as a directed graph G, as described in Section 2. The direction of the edges represents the Hyperonymy relations: $(A, B) \in \{Hyperonymy(X, Y) : X, Y \in NP\}$ represented as $A \rightarrow B$. Let us consider the ('*a salt*', '*a string*') pair. In this case, it denotes the chunk '*the salt is a string*' , which means the *salt* is a specific *string*, a random data in the cryptography used as an additional input for hashing functions, hence the *string* is a generalization of the *salt* in this context.

We have extended the resulted set of edges using the WordNet database. For every source of the edges, we considered the head of that specific noun-phrase as a generalization of that noun-phrase, and these generalizations have been added to the edge list. For these generalized nodes, we have calculated all its generalizations based on WordNet, and these new edges have also been added to the edge list. Using the above example, the head of the source is *salt*, which has many generalizations in WordNet, such as *compound, chemical_compound, flavorer, flavourer, flavoring, flavouring, seasoner, seasoning, taste, taste_sensation, gustatory_sensation, taste_perception, gustatory_perception*. In this case, edges from the node '*salt*' to every element above were added to the list of edges.

The graph representing the semantic network was created from the set of edges using the `networkx` Python module[4]. This graph contains edges extracted from Stack Overflow as well as edges among the nodes of Stack Overflow and nodes of WordNet. The networks has 41,501 nodes and 70,698 edges. Among the nodes there are 5,001 that contain self loops.

Figure 4 presents a small subnetwork with edges representing a connection between Stack Overflow and WordNet. The wealthy meaning of a particular word can be recognized if the different semantic domains are linked to each other. For example the node *salt* can be connected to node *compound* which can be either the *compound key* from the domain of software development, or an *acid* from the domain of chemistry.

The Stack Overflow semantic network links software engineering terms together, sometimes representing deep domain knowledge. On the contrary, WordNet captures the general meaning of terms. The difference of the nature of the captured relations is demonstrated in Table 1. The first column contains the general term, and for each of these we present specific terms from the two different sources in the respective columns. Well known notions in software engineering like defect or hashing are linked to more specific terms compared to WordNet. Evidently, making a connection between the two networks would facilitate the common understanding of communicating parties from different backgrounds.

[4]https://networkx.github.io

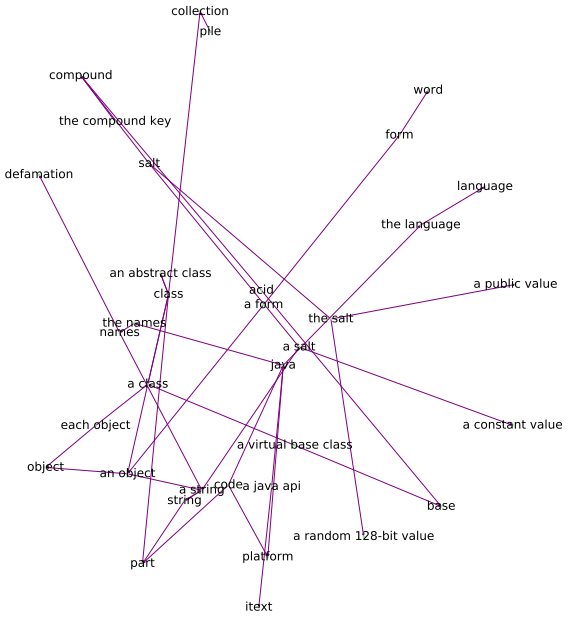

Figure 4: Subgraph of the resulted semantic network

Figure 5 presents the degree distribution of the network. As can be seen, the distribution approximately follows the power-law distribution, which is found in most social networks [6]. The average clustering coefficient is low, 0.0039. The coefficient is calculated as

$$\bar{C} = \frac{\sum \frac{\lambda_G(v)}{\tau_G(v)}}{n},$$

where $\lambda_G(v), v \in NP$ is the number of subgraphs of G with 3 edges (*triangles*) and $\tau_G(v), v \in NP$ is the number of subgraphs. The low coefficient means that the network has only a few cliques.

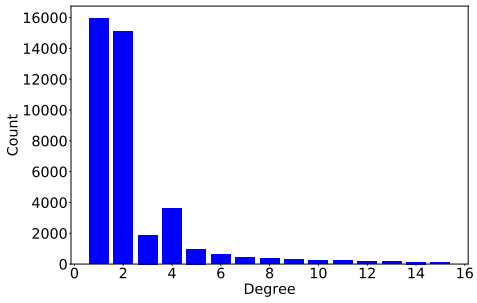

Figure 5: Degree distribution of the network

Table 1: Differences between relations captured by Stack Overflow semantic network and WordNet

General term	Stack Overflow	WordNet
default	consistent across browsers	[delinquency]
weak password hashes	des	[ ]
a defect	a bug	[birth defect, congenital anomaly, congenital defect, congenital disorder, ...]
an accidental complexity	the clunkier syntax	[complicatedness, complication, knottiness, tortuousness, elaborateness, ...]
a one-way operation	hashing	[commission, idle, running, rescue operation, access, memory access, ...]

5 RELATED WORK

Stack Overflow has been the subject of a vast number of research. The quality of the posts, automatic tagging, or the human aspects of the portal are well studied topics [7]. Questions violating the community-established rules will be closed and eventually deleted. Treude et al. [40] investigated whether a question received an accepted answer or remained unanswered. They found that some categories have been answered more frequently than others, like questions from novice developers. Ponzanelli et al. [27] studied the effects of various quality metrics of the questions, which are related to closing, whereas Tóth et al. [39] attempted to examine the quality of the questions based solely on their linguistic features. The automatic tagging also played an essential role in recent studies. Saini et al. [31] and Schuster et al. [32] developed a tag-prediction method based on the words of the question, whereas Beyer et al. [9] applied machine learning techniques to classify posts into seven categories that served as a basis for tag designation.

Applications of semantic networks in the field of software engineering have received more attention recently. These structures have been investigated since the 1980s but software engineering began to exploit the potential of these structures only in the 2000s. Seitner et al. [33] have extracted hypernym relations from the CommonCrawl web corpus. Du and colleagues [12] have developed a tool called OntoSpider to extract ontologies from a web site and convert the pure HTML Web to Semantic Web. Martino et al. [1] conducted a comparison in terms of performance. The quality of NLP result, OWL (*Semantic Web Language*) completeness, and richness between definite-clause formalism and the Watson Relationship Extraction service of IBM Cloud platform Bluemix were studied. Vizcaíno et al. [42] focused on establishing standard vocabularies among the participants of the development, whereas Wongthongtham et al. [44] considered the issues of the multisite developments. In both cases, the aim of the research was to develop a common concept base and consistent information exchange.

Tian et al. [36], Howard et al. [18], Shridhara et al. [35] and Yang and Tan [45] investigated the semantic relationships of the terminologies used in software source code. They found that in the specific semantic area of the Software Engineering, the synonyms between the words are different from the normal usage of those words in English. The authors have worked out methods that can extract these specific synonyms from source codes and the corresponding comments. The results can also help identify the role of the methods used in software, which supports the developers to find a specific function or method during maintenance.

Rashwan et al. [29] applied the Support Vector Machine algorithm for extracting non-functional requirements from SRS. The extraction process is based on ontologies composed by the authors. Balushi et al. [2] developed an ontology-based elicitation tool called ElicitO that helps in the process of capturing precise non-functional requirements (NFRs) specifications during elicitation interviews. Mariza and colleagues [21] applied an ontology-based framework to manage the conflict between usability and security requirements. Kaiya and Saeki [19] used ontologies as domain knowledge to address the completeness issue in requirements engineering. Their early work of modeling the semantic field of a business domain provided proof of concepts in the applicability of the approach in the elicitation process.

6 CONCLUSIONS AND FUTURE WORK

Several methods and frameworks have been created for software engineering to overcome obstacles constituted by communication silos. The main symptom of the difficulty in the communication between two domains is the partly different meanings of the common terms or the usage of those terms that are uniquely assigned to a particular domain. In this work we mined a specific semantic network containing the terms used by software engineers originated from conversations among developers. We used a sample of posts from Stack Overflow to extract one of the most common relationships called hyponym/hyperonym relations. For comparison of the meanings of the terms in different domains, we applied the WordNet dataset, which contains the standard meanings of English words. Using these two datasets, we have demonstrated how to extend the meanings of a specific term, which provides a proof of concept for its usage in software engineering, for example, during the requirements elicitation process.

The extracted dataset can be downloaded from our online repository[5]. In the next step of this research we extend theis dataset by mining the whole Stack Overflow, along with a broader set of lexico-syntactic patterns. We intend to build a semantic network from a different domain to demonstrate the usage based on possible development scenarios.

7 ACKNOWLEDGMENTS

This research was supported in part by the Hungarian Government and the European Regional Development Fund under the grant number GINOP-2.3.2-15-2016-00037 ("Internet of Living Things") and by grant TUDFO/47138-1/2019-ITM of the Ministry for Innovation and Technology, Hungary. László Vidács was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the UNKP-19-4-SZTE New National Excellence Program of the Ministry for Innovation and Technology, Hungary.

[5] https://github.com/sed-inf-u-szeged/SemanticRelations

REFERENCES

[1] 2016. Automatic Production of an Ontology with NLP: Comparison between a Prolog Based Approach and a Cloud Approach Based on Bluemix Watson Service. In *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*. IEEE, 537–542. https://doi.org/10.1109/CISIS.2016.98

[2] Taiseera Hazeem Al Balushi, Pedro R. Falcone Sampaio, Divyesh Dabhi, and Pericles Loucopoulos. 2007. ElicitO: A Quality Ontology-Guided NFR Elicitation Tool. In *Requirements Engineering: Foundation for Software Quality*. Springer Berlin Heidelberg, Berlin, Heidelberg, 306–319. https://doi.org/10.1007/978-3-540-73031-6_23

[3] John R. Anderson and Robert Milson. 1989. Human memory: An adaptive perspective. *Psychological Review* 96, 4 (1989), 703–719. https://doi.org/10.1037/0033-295X.96.4.703

[4] Jeff Atwood. 2018. What does Stack Overflow want to be when it grows up? https://blog.codinghorror.com/

[5] Alan Baddeley, Michael W. Eysenck, and Michael C. Anderson. 2009. *Memory*. Psychology Press, New York, NY, US.

[6] Albert-LÃąszlÃş BarabÃąsi and MÃąrton PÃşsfai. 2016. *Network science*. Cambridge University Press, Cambridge.

[7] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. 2013. On the Personality Traits of StackOverflow Users. In *2013 IEEE International Conference on Software Maintenance*. IEEE, 460–463. https://doi.org/10.1109/ICSM.2013.72

[8] Christine A. Bevc, Jessica H. Retrum, and Danielle M. Varda. 2015. New perspectives on the "silo effect": initial comparisons of network structures across public health collaboratives. *American journal of public health* 105, Suppl (2015). https://doi.org/10.2105/AJPH.2014.302256

[9] Stefanie Beyer, Christian Macho, Martin Pinzger, and Massimiliano Di Penta. 2018. Automatically classifying posts into question categories on stack overflow. In *Proceedings of the 26th Conference on Program Comprehension - ICPC '18*. ACM Press, New York, New York, USA, 211–221. https://doi.org/10.1145/3196321.3196333

[10] Denzil Correa and Ashish Sureka. 2013. Fit or Unfit: Analysis and Prediction of 'Closed Questions' on Stack Overflow. In *Proceedings of the First ACM Conference on Online Social Networks (COSN '13)*. ACM, New York, NY, USA, 201–212. https://doi.org/10.1145/2512938.2512954

[11] Richard L. Tenney Dan A. Simovici. 1999. *Theory of Formal Languages with Applications*. 629 pages.

[12] Timon C. Du, Feng Li, and Irwin King. 2009. Managing knowledge on the Web Extracting ontology from HTML Web. *Decision Support Systems* 47, 4 (nov 2009), 319–331. https://doi.org/10.1016/J.DSS.2009.02.011

[13] Daniel E. 2018. The Silo Effect | Case Study On The Silo Effect The Strategic CFO. https://strategiccfo.com/silo-effect/

[14] Donald Firesmith. 2007. Common requirements problems, their negative consequences, and the industry best practices to help solve them. *Journal of Object Technology* 6, 1 (2007), 17–33. https://doi.org/10.5381/jot.2007.6.1.c2

[15] Brent Gleeson. 2013. The Silo Mentality: How To Break Down The Barriers. https://www.forbes.com/sites/brentgleeson/2013/10/02/the-silo-mentality-how-to-break-down-the-barriers/

[16] Prahlad Gupta. 2009. A computational model of nonword repetition, immediate serial recall, and nonword learning. In *Interactions between short-term and long-term memory in the verbal domain*. Psychology Press, New York, NY, US, 108–135.

[17] Marti A Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *{COLING} 1992 Volume 2: The 15th International Conference on Computational Linguistics*. https://www.aclweb.org/anthology/C92-2082

[18] M. J. Howard, S. Gupta, L. Pollock, and K. Vijay-Shanker. 2013. Automatically mining software-based, semantically-similar words from comment-code mappings. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. 377–386. https://doi.org/10.1109/MSR.2013.6624052

[19] H. Kaiya and M. Saeki. 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation. *14th IEEE International Requirements Engineering Conference (RE'06)* (2006), 189–198. https://doi.org/10.1109/RE.2006.72

[20] Peter Rolf Lutzeier. 1982. The notion of lexical field and its application to english nouns of financial income. *Lingua* 56, 1 (1982), 1–42. https://doi.org/10.1016/0024-3841(82)90048-1

[21] Dewi Mairiza and Didar Zowghi. 2010. An ontological framework to manage the relative conflicts between security and usability requirements. In *2010 3rd International Workshop on Managing Requirements Knowledge, MaRK'10*. https://doi.org/10.1109/MARK.2010.5623814

[22] Macarena Navarro-Pablo. 1999. The lexical field, a key to semantics. *Cauce: Revista Internacional de Filología, Comunicación y sus Didácticas* 22 (1999), 539–548. https://dialnet.unirioja.es/servlet/articulo?codigo=623228

[23] ADAM PEASE and IAN NILES. 2002. IEEE standard upper ontology: a progress report. *The Knowledge Engineering Review* 17, 1 (mar 2002), 65–70. https://doi.org/10.1017/S0269888902000395

[24] Csaba Pléh and Ágnes Lukács. 2009. *Pszicholingvisztika*. Akadémiai kiadó Zrt. 1484 pages.

[25] Marie Poirier, Jean Saint-Aubin, Ali Mair, Gerry Tehan, and Anne Tolan. 2015. Order recall in verbal short-term memory: The role of semantic networks. *Mem Cognit* 43, 3 (apr 2015), 489–499. https://doi.org/10.3758/s13421-014-0470-6

[26] Roberto Poli, Michael Healy, and Achilles Kameas (Eds.). 2010. *Theory and Applications of Ontology: Computer Applications*. Springer Netherlands. https://doi.org/10.1007/978-90-481-8847-5

[27] Luca Ponzanelli, Andrea Mocci, Alberto Bacchelli, and Michele Lanza. 2014. Understanding and Classifying the Quality of Technical Forum Questions. In *2014 14th International Conference on Quality Software*. IEEE, 343–352. https://doi.org/10.1109/QSIC.2014.27

[28] Princeton University. 2010. About WordNet. https://wordnet.princeton.edu/

[29] Abderahman Rashwan, Olga Ormandjieva, and Rene Witte. 2013. Ontology-based classification of non-functional requirements in software specifications: A new corpus and SVM-based classifier. In *Proceedings - International Computer Software and Applications Conference*. IEEE, 381–386. https://doi.org/10.1109/COMPSAC.2013.64

[30] IA Richards and CK Ogden. 1989. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Harvest/HBJ.

[31] Taniya Saini and Sachin Tripathi. 2018. Predicting tags for stack overflow questions using different classifiers. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 1–5. https://doi.org/10.1109/RAIT.2018.8389059

[32] Sebastian Schuster, Wanying Zhu, and Yiying Cheng. 2017. Predicting Tags for StackOverflow Questions. In *Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB*.

[33] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A Large DataBase of Hypernymy Relations Extracted from the Web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation {LREC} 2016, Portorož, Slovenia, May 23-28, 2016*, Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association {(ELRA)}. http://www.lrec-conf.org/proceedings/lrec2016/summaries/204.html

[34] John F. Sowa. 1987. *Semantic Networks*.

[35] G. Sridhara, E. Hill, L. Pollock, and K. Vijay-Shanker. 2008. Identifying Word Relations in Software: A Comparative Study of Semantic Similarity Tools. In *2008 16th IEEE International Conference on Program Comprehension*. 123–132. https://doi.org/10.1109/ICPC.2008.18

[36] Y. Tian, D. Lo, and J. Lawall. 2014. Automated construction of a software-specific word similarity database. In *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*. 44–53. https://doi.org/10.1109/CSMR-WCRE.2014.6747213

[37] De Smedt Tom and Daelemans W. 2012. Pattern for Python. *Journal of Machine Learning Research* 6 (2012), 2031–2035.

[38] László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, and Tibor Gyimothy. 2019. Towards an Accurate Prediction of the Question Quality at Stack Overflow Using a Deep-Learning-Based NLP Approach. In *Proceedings of ICSOFT 2019, 14th International Conference on Software Technologies*. 631–639. https://doi.org/10.5220/0007971306310639

[39] László Tóth and László Vidács. 2019. Comparative Study of The Performance of Various Classifiers in Labeling Non-Functional Requirements. *Information Technology and Control* 48, 3 (2019), 432–445. https://doi.org/10.5755/j01.itc.48.3.21973

[40] Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey. 2011. How do programmers ask and answer questions on the web?. In *Proceeding of the 33rd international conference on Software engineering - ICSE '11*. ACM Press, New York, New York, USA, 804. https://doi.org/10.1145/1985793.1985907

[41] Hossein Vatanpour, Atoosa Khorramnia, and Naghmeh Forutan. 2013. Silo Effect a Prominence Factor to Decrease Efficiency of Pharmaceutical Industry. *Iran J Pharm Res* 12, Suppl (2013), 207–216. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3813367/

[42] Aurora Vizcaíno, Felix García, Mario Piattini, and Sarah Beecham. 2016. A validated ontology for global software development. *Computer Standards & Interfaces* 46, C (may 2016), 66–78. https://doi.org/10.1016/j.csi.2016.02.004

[43] Xiaohua Wang, Zhi Wu, and Ming Zhao. 2008. The Relationship between Developers and Customers in Agile Methodology. In *2008 International Conference on Computer Science and Information Technology*. 566–572. https://doi.org/10.1109/ICCSIT.2008.9 ISSN: null.

[44] P. Wongthongtham, E. Chang, T. Dillon, and I. Sommerville. 2009. Development of a Software Engineering Ontology for Multisite Software Development. *IEEE Transactions on Knowledge and Data Engineering* 21, 8 (aug 2009), 1205–1217. https://doi.org/10.1109/TKDE.2008.209

[45] J. Yang and L. Tan. 2014. SWordNet: Inferring semantically related words from software context. In *Empirical Software Engineering*. 1856–1886. https://doi.org/10.1007/s10664-013-9264-x