Natural Language Processing and Artificial Intelligence Methods in Software Engineering

PhD Thesis

László Tóth

Supervisors

Tibor Gyimóthy, DsC
László Vidács, PhD

Doctoral School of Computer Science
Department of Software Engineering
Faculty of Science and Informatics
University of Szeged

Szeged
February 2022

1 Introduction

The proliferation of programmable devices entailed the need for increasingly complex software accountable for the operation of these devices. Besides, the increasing performance of computers has also allowed traditional software to implement an ever-wider spectrum of functionality. The boost in computing power and the development of artificial intelligence have made it possible to solve tasks that only humans could previously do.

The key to successful software development lies in the availability of reasonable quality requirements with a proper level of detail. The requirements are the rules and conventions defined by the technical, economic, and social environment, expressed in codified business rules, legislation, and technical documents or provided by the stakeholders using a natural language. In the latter case, requirements can be incomplete, inaccurate, or have contradictions due to the different needs of the stakeholders.

Human communication is often fraught with difficulties for several reasons. One of the most common of these is the language usage determined by different cultural and professional backgrounds, which interprets into different meanings expressed in the same terms. These communication difficulties play a significant role in software development, as successful development and customer support require customers and developers to understand the same when communicating.

As software has been used to focus on solving increasingly complex problems, ensuring the reusability of code developed for repetitive tasks has become an essential consideration for software developers. Reusability has also led to a rapid increase in development libraries for programming languages. At the same time, various frameworks have been developed, increasing the efficiency of development. With the growth in developer tools, programmers can no longer keep up, so developers need to specialize in a particular field, programming language, or framework. This specialization requires increased collaboration such developers that is also supported by various Q&A portals on the internet. The best known of these portals is Stack Overflow, founded in 2008 by Jeff Atwood and Joel Spolski [8] to provide developers with professional support to solve the problems they face in their daily work. In order to maintain professionalism and quality, Stack Overflow has adopted a set of rules that are sometimes challenging to follow and check, both for moderators and users.

The Ph.D. thesis presents research supporting software engineering processes using natural language processing and artificial intelligence methods. The research focused on the requirement engineering processes and the communication between developers using Stack Overflow in their communication.

The dissertation consists of three major parts. The first chapter introduces the topic selection and the motivation. The second chapter presents research supporting the requirements analysis processes using natural language processing and artificial intelligence methods. The requirements given in natural languages can be classified into functional and non-functional requirements, where the last are often overlooked. Based on the classification results, the non-functional requirements can be collected and handled correctly during the analysis and software design. The other research flow of this topic aims to support the communication between the business and software engineering domain using semantic networks to resolve the difficulties caused by different meanings of the same terms used in the communication process.

The third part of the dissertation presents research analyzing the interactions between software developers using natural language processing and deep learning methods. The research focused on the questions developers asked on Stack Overflow, particularly the quality of those questions and the likelihood of the closure, along with the possible reason. The result of the research can be utilized by incorporating the models into a tool that can help both the moderators and the questioners with checking the questions, whether (or not) the question meets the expectations of the portal, thus reducing the likelihood of closures.

2 Natural Language Processing and Artificial Intelligence Methods in Requirement Analysis

The quality of requirements is the primary factor in a software project's success. The lack of a well-structured set of non-functional requirements can lead to an inappropriate software design and the project's failure. Many NFRs (Non Functional Requirements) are out of the analysis, and those considered during analysis are often weakly elaborated. Firesmith, in his article issued in the Journal of Object Technology in 2007, has collected the most common issues related to requirements engineering along with some practice to solve these problems [9]. Although steps have been taken to improve the quality of the specifications, unsuccessful software projects are still being attributed mainly to inadequate requirements engineering [10].

The duty of the business analysts is to organize the requirements provided by different stakeholders and create a formal or semi-formal model suitable for software design for the development team. In order to perform this task at the appropriate level, business analysts need to have a thorough understanding of the requirements and their implications. The principal difficulty of this task is the usage of the same terminologies with different meanings and contexts along with the related tacit knowledge of the business side, which is usually not articulated during the elicitation process.

The problem mentioned above lies in the specific nature of human communication. Information exchange is influenced by several factors, such as cultural background, social environment, the available communication channels, personality and mental state of the participants, and their communication intention, even when a common language is used. The cultural background and the social environment have paramount importance because they can affect the actual meanings of the words used in the communication. This phenomenon, called the communication silo [11], is increasingly present in communication between IT and the business area [12].

Mapping the different semantic fields provides a possible solution for reconciling the different meanings and catching the corresponding tacit knowledge. The notions used in a particular domain often provide another or overplus meaning to the words denoting them. Semantic networks [13] are proper tools representing the semantic relationships between the terms used in a particular domain, and matching them together can support recognizing the different meanings and catch the tacit knowledge among the participants of different domains.

Chapter 2 of the Ph.D. thesis presents the research related to the requirement analysis covering both the recognition and classification of non-functional requirements and supporting the resolution of the communication barriers originating from the language usage of the different semantic fields related to software engineering.

2.1 Classification of Non-Functional Requirements

Requirements in a textual form are to be adequately preprocessed and transformed into a vectorized form that computers can process more efficiently. The procedure involves standard natural language techniques such as tokenization and filtering. For vectorization, the *tf-idf* model is one of the most common representations, also used in the recent research [14], which was used in our classification experiments.

Two different datasets were involved in our experiments. For the first, the Tera Promise NFR dataset was used, which was constructed by the MS students of the DePaul University [15]. The source of the second experiment was a dataset queried from Stack Overflow. The later dataset comprises a sequence of tagged posts containing English texts, and often code fragments also occur. We have chosen posts tagged with *performance* or *test-related* labels for the experiment. *Testability* and *Performance* are crucial factors for software quality, and these topics are also discussed thoroughly via Stack Overflow. During the preprocessing phase, the labels of the dataset from Stack Overflow were also transformed using *one-hot encoding*, and the labels different from our interests were filtered out.

In the first experiment, we performed classifications using algorithms implemented in the *scikit-learn* library on the dataset of Promise NFR and compared the results with each other and with the results obtained by Cassamayor et al. [16]. The objective of this experiment was to determine the best classification algorithm implemented in the *scikit-learn* library for requirements classification tasks.

Table 1: *Precision, recall, and F-measure of the classification of the Promise NFR*

Classifier	Average			Comp	Variance		
	P	R	F	F	P	R	F
BernoulliNB	0.43	0.22	0.25	0.29	0.18	0.09	0.06
DT	0.66	0.64	0.62	0.65	0.01	0.03	0.01
ET	0.63	0.62	0.59	0.62	0.01	0.04	0.02
ETs	0.63	0.63	0.59	0.63	0.01	0.04	0.02
GNB	0.72	0.69	0.67	0.70	0.02	0.02	0.02
KNeighbours	0.71	0.52	0.55	0.60	0.08	0.07	0.05
LabelPropagation	0.70	0.68	0.65	0.69	0.02	0.03	0.02
LabelSpread	0.70	0.67	0.65	0.68	0.02	0.03	0.02
Logistic	0.87	0.67	0.72	0.76	0.01	0.05	0.03
MLP	0.38	0.66	0.36	0.48	0.01	0.03	0.01
MultinomialNB	0.84	0.68	0.72	0.75	0.02	0.03	0.02
SVM	0.89	0.65	0.71	0.75	0.01	0.05	0.02

In Table 1, the measured averages of Precision, Recall, and F-measure related to the experiment on the Tera Promise NFR dataset with their averaged variance are presented. The results show that the SVM has produced the best precision value; however, the recall is only 65% which is the median value of the results. The Multinomial Naive Bayes and

Logistic Regression have produced the best F1 values. As one can see in the table, the MLP has produced the worst result. The dataset size is too small for applying the Multilayer Perceptron classifier, and this result can be explained with the underfitted model.

In the Stack Overflow-based experiment, models implemented in the `scikit-learn` library were supplemented with a Tensorflow- and Keras-based Fully Connected Neural Network implementation. The MLP (Multilayer Perceptron) model implemented in the `scikit-learn` was removed from these experiments because the Tensorflow implementation provides suitable scalability, and the applied strategy is similar in the two cases. The Label Propagation and the Label Spreading models were also removed because a performance issue was detected applying them on the Stack Overflow dataset.

Table 2: *Precision, recall, and F1 values on the Stack Overflow dataset*

Classifier	micro average			macro average		
	P	R	F	P	R	F
BernoulliNB	0.91	0.91	0.91	0.90	0.90	0.90
GaussianNB	0.82	0.81	0.81	0.84	0.84	0.81
MultinomialNB	0.92	0.91	0.92	0.92	0.92	0.92
DecisionTree	0.91	0.91	0.91	0.91	0.90	0.90
ExtraTree	0.84	0.84	0.84	0.83	0.83	0.83
LogisticRegr	0.95	0.95	0.95	0.95	0.95	0.95
KNeighbours	0.80	0.80	0.80	0.82	0.76	0.77
SVM	0.95	0.94	0.94	0.94	0.94	0.94
FullyConnected	0.96	0.94	0.95	0.96	0.94	0.95

In the experiment with the Stack Overflow posts, the metrics were calculated using classification reports provided by the *scikit-learn* library. The report used those calculation methods applied in the experiment with the Promise NFR dataset; however, the averaged measures were calculated manually for the Promise NFR dataset, whereas in the case of Stack Overflow experiments, the classification report produced the average values.

Table 3: *Precision, recall, F1 values on the Stack Overflow dataset*

Classifier	weighted average			samples average		
	P	R	F	P	R	F
BernoulliNB	0.91	0.91	0.91	0.91	0.91	0.91
GaussianNB	0.87	0.81	0.81	0.82	0.82	0.81
MultinomialNB	0.92	0.91	0.92	0.93	0.92	0.92
DecisionTree	0.91	0.91	0.91	0.90	0.91	0.90
ExtraTree	0.84	0.84	0.84	0.80	0.85	0.80
LogisticRegr	0.95	0.95	0.95	0.95	0.96	0.95
KNeighbours	0.80	0.80	0.79	0.80	0.80	0.80
SVM	0.95	0.95	0.94	0.94	0.95	0.94
FullyConnected	0.96	0.94	0.95	0.96	0.95	0.95

The results are presented in Tables 2 and 3, respectively. According to these results, the Logistic Regression, the SVM, and the FCN (Fully Connected Network) have produced the best values. The FCN has yielded the highest values of precision and F-measure in all calculated averages.

The results of the experiments on the Stack Overflow dataset confirmed the results on the Tera Promise NFR, and the following theses can be established:

- Based on the references to non-functional requirements in the text, requirements can be efficiently classified using linear models based on `tf-idf` vectorization. Based on the experiments, the best performance was achieved by the following three models: SVM, Multinomial Naive Bayes, and Logistic Regression.
- In the case of a significant number of learning examples, the neural network models classify non-functional requirements more efficiently than the traditional linear models.

2.2 Mining Hyperonym Relations from Stack Overflow

The most frequent issues of software projects are requirements comprehension and establishing 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.

The meaning of an abstract concept in a given context marked by a particular term is determined by its relation to other concepts valid in the same context. Processing concepts marked with their terms by a computer program requires awareness of these relationships, which, in turn, can help clarify the actual meaning of a given term. The relationship examined in this research is a model of super-subordinate relations or, as called in linguistics, hyperonym-hyponym relations. This bond is also typical in object-oriented analysis and design, the generalization-specification relationship, also called inheritance.

Definition 1 *Semantic network is a*

$$G = (C, R, \Sigma_1, \Sigma_2, s, d, l_1, l_2)$$

labeled directed multigraph, where C is the set of nodes representing the concepts of a given domain, and R is the set of edges representing the relationships between the elements of C. Σ_1 and Σ_2 are the alphabets of the labels corresponding to the nodes and edges, respectively. The $s, d : R \to C$ are the source and destination functions:

$$\forall r \in R,\ \exists (X, Y \in C) : r = (X, Y) \wedge X = s(r), r \wedge Y = d(r).$$

Similarly, $l_1 : \Sigma_1 \to C$ and $l_2 : \Sigma_2 \to R$ are the two labeling functions for the nodes and the edges, respectively.

The connection between the world and the concepts has a third component, without which communication would be impossible. This part is the marker associated with concepts, most often a linguistic phrase. This marker is called a term or, in other modalities, a symbol. Ideally, the relationship between concepts and their linguistic markers in a given language would be injective, but in reality, this is not the case. Injection only exists in a narrower context called *semantic space*.

Definition 2 *Let C be an n-element set of concepts and $C_k \subseteq C$ its k-element subset $(k \leq n)$. Let*

$$S = (c_1, f_1), (c_2, f_2), ..., (c_k, f_k)$$

$c_i \in C_k\ ,\ f_i \in F(i = 1, ..., k \leq n)$ *be a series, where*

$$F := \{f | f : C \times C_k \to [0, 1]\}$$

is a set of membership functions designating those $c_j \in C_k$ $(j = 1, ..., k)$ concepts that participate in defining a particular $c_i \in C$ concept $(i = 1, .., .n)$. Let $V = \boldsymbol{R}^k$ be a vector space over $\boldsymbol{R}$ and $\Psi : C \to V$ surjective mapping as follows:

1. $\forall c \in C : \Psi(c) = \boldsymbol{v}(v_1, v_2, ..., v_k)\ \in V$.
2. $v_i = f_i(c, c_i) \in [0, 1]$, $(i = 1, ..., k)$*, where* $(c_i, f_i) \in S$ *and* $(c, c_i) \in C \times C_k$*, a fuzzy relationship between* c *and* c_i, $(i = 1, ..., k)$*.*

Given S, the vector space V is called the semantic space of C if the mapping Ψ is bijective.

Semantic networks restricted to various semantic spaces provide a comprehensive and tractable way for examining the semantic relationships among the concepts denoted by various word phrases. These relationships can be defined in various ways. We focus on the generalization and specialization, which is called *hypernymy* and *hyponymy* relations in linguistics.

Definition 3 *Let C be the set of concepts and let L be a natural language with the alphabet Σ. Let Σ^* be the constraint of the set of words over L for the valid words, and word phrases of L. Let $X, Y \in \Sigma^*$, and let $M : \Sigma^* \to C$ be a mapping from the word phrases to the set of concepts. Let P the set of properties describing the objects of the world, and let $\Pi : C \to P$ a mapping from the set of concepts to the set of properties. Hyponymy relation between X and Y is defined as*

$$Hyponym(X, Y) \iff \Pi(M(Y)) \subset \Pi(M(X)).$$

Hyperonymy relation, denoted as $Hyperonym(X, Y)$ is the inverse relation of the hyponymy.

Stack Overflow is an extensive knowledge repository in software development and engineering. The phrases found in its posts might serve as a base to build a semantic network for the software development domain. Nevertheless, some critical issues need to be considered upon using this dataset. The posts often contain code fragments or unique character strings used in programming. Besides, many posts are written by non-English speakers; therefore, they might contain smaller or bigger grammatical errors.

In terms of content value, tacit knowledge in Stack Overflow posts can be trusted to be of high quality because of the strict community control of the portal. Posts that fail to meet requirements established by the community are to be closed and eventually deleted. SO posts, therefore, are a good source for modeling the semantic domain of software development. We used posts in our experiments from the Stack Overflow data dump created on March 4, 2019.

Hyponymy and hyperonymy relations can be extracted from free texts using *lexico-syntactic patterns* proposed by Hearst [17]. These relationships represent the canonical *"is-a"* relationship, which can also be interpreted as the specialization and the generalization in object-oriented modeling. Since Hearst established the base models, the collection of *lexico-syntactic patterns* has been expanded. The momentum of this expansion comes from the rapid spread of web-based text mining. Our work adopted the patterns introduced in the article of Seitner et al. [18] The patterns used in our research were used in a slightly modified form, aiming to avoid confusion with extracting noun phrases from the text and writing more compact code.

The objective of the preprocessing phase is to provide a cleaned input text sliced into sentences for the mining process. The raw text contains elements that have to be eliminated or simplified. Besides, it has to be ensured that the processed text contains only relevant and accepted questions and answers. Therefore, only posts that obtained non-negative scores from the Stack Overflow community were considered. Code blocks and hyperlinks from the text were removed and replaced with special strings. The HTML tags were removed, and special characters were replaced with whitespace characters, but the punctuation characters were retained. The text was then split into sentences.

In computer-based language processing, we need to formally define the examined linguistic structures, which approximates the set of the structures used in reality. For this purpose, the phrase structure grammar is a suitable choice due to its algorithmic manageability. For matching *lexico-syntactic patterns* to the input text, *identifying noun phrases* in the original text is required.

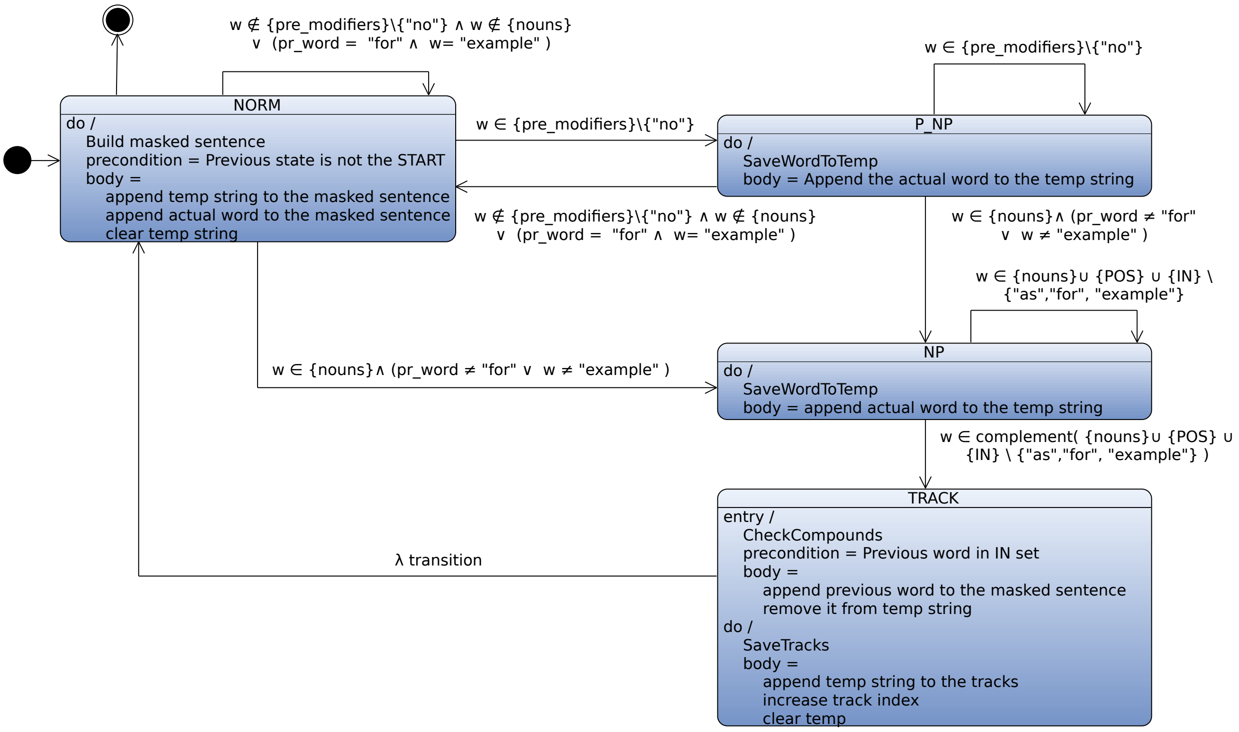

w ∉ {pre_modifiers}\{"no"} ∧ w ∉ {nouns}
∨ (pr_word = "for" ∧ w= "example")
w ∈ {pre_modifiers}\{"no"}
NORM
do /
Build masked sentence
precondition = Previous state is not the START
body =
append temp string to the masked sentence
append actual word to the masked sentence
clear temp string
w ∈ {pre_modifiers}\{"no"}
P_NP
do /
SaveWordToTemp
body = Append the actual word to the temp string
w ∉ {pre_modifiers}\{"no"} ∧ w ∉ {nouns}
∨ (pr_word = "for" ∧ w= "example")
w ∈ {nouns}∧ (pr_word ≠ "for"
∨ w ≠ "example")
w ∈ {nouns}∪ {POS} ∪ {IN} \
{"as","for", "example"}
w ∈ {nouns}∧ (pr_word ≠ "for" ∨ w ≠ "example")
NP
do /
SaveWordToTemp
body = append actual word to the temp string
w ∈ complement({nouns}∪ {POS} ∪
{IN} \ {"as","for", "example"})
λ transition
TRACK
entry /
CheckCompounds
precondition = Previous word in IN set
body =
append previous word to the masked sentence
remove it from temp string
do /
SaveTracks
body =
append temp string to the tracks
increase track index
clear temp

Figure 1: *Extracting noun phrases*

The mining process and the experiments were conducted twice. The first experiment examined the applicability of using semantic networks for collating different meanings of terms applied in the different semantic domains. For this purpose, a sample of questions containing 1.3 M sentences was selected randomly from the original dataset after the preprocessing phase. In this experiment, we applied simplified automation for recognizing noun phrases, as shown in Figure 1. The simplified automation could be used as we only selected the two most frequent lexico-syntactic patterns to extract the relationships:

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

NP_t denotes the hyponym part of the pattern, whereas NP_h means the hypernym member. The parentheses indicate optional components of the pattern, and pipe (|) signifies a choice among more than one constituent.

The graph obtained from Stack Overflow posts was compared with the WordNet [19], an extensive lexical-semantic database of English, for testing its applicability in practice.

Table 4: *Differences between relations in Stack Overflow 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, ...]

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 captured relations is demonstrated in Table 4. 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 a *defect* or *hashing* are linked to more specific terms than WordNet. Making connections between the two networks would facilitate the common understanding of communicating with parties from different backgrounds.

After testing the concept, a more complete graph was built on the whole dataset, using a larger set of lexico-syntactic patterns and a precise and detailed definition of the grammar and automation that recognizes noun phrases. For this purpose, a phrase structure grammar was developed, which is shown in Figure 2. The NP parser is based on the automation presented in Figure 3. The automation utilizes the POS tags to compute the proper transition. Although the recognition of NPs and lexico-syntactic patterns is a separate procedure, the lexical elements *(e.g.: some, any, kind, sort, etc.)* used in lexico-syntactic patterns should be considered when recognizing NPs.

The lexico-syntactic patterns used in the extraction process assume an input text written with proper English grammar. However, Stack Overflow posts are often written by non-English speaking users who sometimes make grammatical errors. These errors might result in a wrong relation extracted from the text. Therefore, a few post-processing steps have been introduced to reduce the number of mismatched pairs.

After the post-processing phase, the extracted pairs are ready for graph building. These pairs provide the set of edges of the semantic graph. Unfortunately, duplications can also

$$\begin{array}{rcl}
\langle \text{NP} \rangle & \models & (\text{PDT})\ (\text{DET}\ (\text{CD})\ \mid\ \text{PRP\$}) \\
 & & ((\langle \text{ADJPS} \rangle)\ \langle \text{HEAD} \rangle\ ((\langle \text{GERS} \rangle)\ ((\langle \text{PPS} \rangle)\ ((\langle \text{REL} \rangle) \\
\langle \text{ADJPS} \rangle & \models & ((\langle \text{ADVS} \rangle)\ \langle \text{ADJ} \rangle\ ((\langle \text{CONJ} \rangle\ \langle \text{ADJ} \rangle)^{*} \\
\langle \text{ADVS} \rangle & \models & \langle \text{ADV} \rangle\ ((\langle \text{CONJ} \rangle\ \langle \text{ADV} \rangle)^{*} \\
\langle \text{ADJ} \rangle & \models & \text{JJ}\ \mid\ \text{JJR}\ \mid\ \text{JJS} \\
\langle \text{ADV} \rangle & \models & \text{RB}\ \mid\ \text{RBR}\ \mid\ \text{RBS} \\
\langle \text{HEAD} \rangle & \models & \text{NN(POS)}\ \mid\ \text{NNP(POS)}\ \mid\ \text{NNPS(POS)}\ \mid \\
 & & \text{NNS(POS)}\ \mid\ \text{PRP}\ \mid\ \text{SYM}\ \mid\ \text{FW} \\
\langle \text{GERS} \rangle & \models & \langle \text{GER} \rangle\ ((\langle \text{CONJ} \rangle\ \langle \text{GER} \rangle)^{*} \\
\langle \text{GER} \rangle & \models & \text{VBG}\ ((\langle \text{NP} \rangle) \\
\langle \text{PPS} \rangle & \models & \text{IN}\ \langle \text{NP} \rangle\ \mid\ \text{TO VB}\ ((\langle \text{CONJ} \rangle\ \text{VB})^{*} \\
\langle \text{CONJ} \rangle & \models & ,\ \mid\ \text{CC} \\
\langle \text{REL} \rangle & \models & \text{WDT}\ \mid\ \text{WP}\ \mid\ \text{WP\$}\ \mid\ \text{WRB}\ \langle \text{VP} \rangle
\end{array}$$

Figure 2: *Grammar defined for NP recognition*

occur among these pairs, which must be removed during the graph building process. For building the graph, the `networkx` Python package was utilized. The resulting graph was then imported into a Neo4J database.

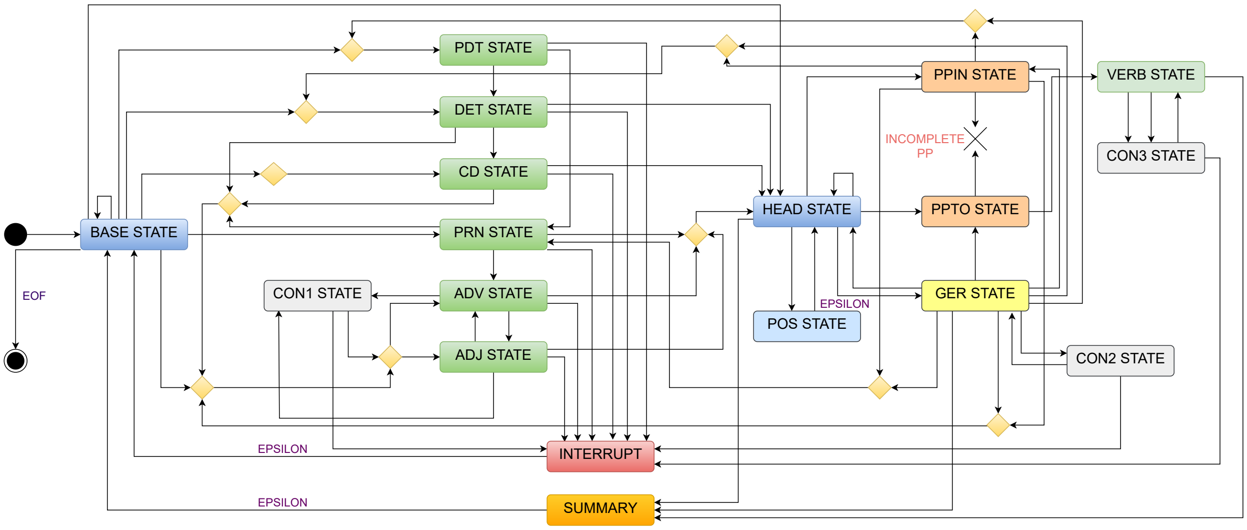


Figure 3: *NP Extractor Automation*

After the extraction process, the distribution of the applied patterns was examined. According to the distribution, the following three lexico-syntactic patterns are used the most dominantly in Stack Overflow posts: $NP_t\ is|are|was|were\ (a|an)\ NP_h$, $NP_t\ as\ NP_h$ and $NP_h\ like\ NP_t$. The occurrence of the following patterns is marginal: $NP_t\ or\ the\ many\ NP_h$ and $NP_h\ example\ of\ this\ is|are\ NP_t$. The statistical results were compared to those of Seitner et al. [18] They applied a similar lexico-syntactic pattern-based mining on the dataset obtained from CommonCrawl using a slightly different grammar for NP identification and, therefore, a slightly different set of patterns. Despite the differences, we found that our patterns followed a similar trend.

The degree distribution of the resulting graph was also examined. The result of the analysis shows that in the case of the hyponymy → hyperonymy direction, the distribution follows the *Box-Cox Power Exponential Distribution*, whereas in the opposite direction, the distribution follows the *Power-Law Distribution*.

The average clustering coefficient is 0.017, meaning that only 1.7% of the concepts tend to form triadic closures. A possible explanation for this small number can be that large degree nodes connect the communities formed in the knowledge graph. This phenomenon can be interpreted as the concepts being defined based on a few core concepts. Further studies are needed to confirm this conjecture.

Based on our investigations, the following theses can be established:

- Based on the examination of the related literature and the investigation of the semantic relationships of the terms used in a specific and common semantic domain, it is confirmed that the meaning of the terms used in communication may differ significantly in the different semantic domains.
- Semantic networks are appropriate tools to resolve the confusion originating from the different meanings of the same terms.
- The semantic network that presents the hyperonym-hyponymy relationships based on the Stack Overflow posts has a specific structure. It is a directed graph; the degree distribution follows a Box-Cox Exponential Distribution in the hyponymy → hyperonymy direction and the Power Law Distribution in the opposite direction, respectively. Besides, the graph has a very low clustering coefficient, meaning that only a few concepts tend to form triadic closures.

3 Investigating Developers Interactions Using Natural Language Processing And Deep Learning

Stack Overflow (SO) is probably the most influential community question answering site for the software engineering and development society. With more than 11 million registered users and over 6,500 new questions posted on the site every day, SO has undoubtedly become an essential technical knowledge repository in programming. As the community grew in size and the less experienced users became more active on the site, the moderators' duties became increasingly demanding. However, the community remained determined to maintain the professionalism and quality of the site and often closed issues that did not meet the requirements established by the community [20]. At the same time, less experienced users found it increasingly difficult to adapt to the strict quality requirements, which can be daunting for newcomers, and increasingly encountered the closure of questions without response, which they experienced as frustration. The reason for closing questions is not always apparent, which has led to disapproval not only from novices but also the proficient users [21].

Questions posted on the portal should be relevant, unambiguous, and comprehensible, meaning that every question has to be related either to specific development issues or methods such as using a particular API, algorithmic problems, tools, or methodology. Questions that cannot be answered suitably and straightforwardly can face closure and eventual deletion. Currently, there are five reasons for closing a question: due to *duplication*, the question is *off-topic*, *unclear* what the user is asking, the question is *too broad* and cannot be answered straightforwardly, the question is primarily *opinion-based* leading to subjective discussions.

The closing procedure is a manual task relying on a voting system. Moderators or privileged users with 3,000 or more reputation points can cast a vote if they find a question inappropriate, and five such votes would close the question. A previously closed question can be deleted if it receives at least three votes from members within the 10,000 or higher reputation range. Reputation points and different badges reflect both the experience and the activity of the users on Stack Overflow.

Chapter 3 of the Ph.D. thesis presents our research about predicting the quality of the questions, their likelihood for closing, and possible closing reasons. Based on the research results, a tool can be constructed that helps both the moderators and the users evaluate the likelihood of closing a given question leaning on only the textual information known during the assembly of that question.

3.1 Quality based experiments

The dataset for this study was taken from the SO data dump created on March 4, 2019. Both open and closed questions posted before June 2013 – the introduction of the currently available closing policy and reasons – were filtered. Questions closed due to duplication were also excluded from our dataset because of two reasons: *i*) this problem has already been well examined previously, and *ii*) finding a duplicate would imply the knowledge of the previously posted questions.

The first study focused on the classification of the quality of the questions. In defining high and low quality, we relied on the definition given by Ponzanelli et al. [22], which allowed us to compare our results with theirs. We consider those questions high quality that have at least one accepted answer, their score is higher than zero, and they are neither closed nor deleted. Those questions that own a score less than zero and closed or deleted in their final states were considered low-quality.

We have to highlight that neither the online database nor the dump contain permanently deleted questions. Those data can be obtained only by the moderators. This fact has a detrimental effect on the models used; the smaller set of training examples can cause the models to perform worse than expected.

Based on the quality definition and the distribution obtained from the online database, a significant number of questions cannot be classified mainly due to the vast amount of questions that have scored equal to zero, and even questions with positive scores do not have any accepted answers.

The results of the quality-based experiments were intended to compare with the results of the model of Ponzanellis' who had used the dump of the Stack Overflow created in 2013 September. That dump is not available anymore; therefore, we have selected the corresponding period from our dataset for the quality-based classifier. The test dataset was compiled using balanced subsets of high-quality and low-quality questions. Our objective was to decide on the potential rejection regarding a question to be posted to Stack Overflow using only the linguistic features; therefore, the training and the test set contain only the question-body, the title, and the tags attached to the given post. The quality class and the Id of the posts were also included. Four separate experiments were executed using two different vectorization processes.

For the first three experiments, the *Spacy* library was used to calculate the vector representing the related posts. The difference between the experiments is the number of training examples *(for comparison purposes with the Ponzanelli's setups)* and the calculated metrics containing cosine similarity between the parts of the input vector in the case of the third experiment (1.3). For the fourth experiment, a 200-dimensional *Doc2Vec* model on Stack Overflow questions using the *Gensim* library was constructed. This model is based on the *Word2Vec* model with the addition of the feature vector related to the whole document. Besides, the more expanded dataset was involved as the training set in the experiment. The common preprocessing steps, such as text cleaning or tokenization, were also performed before the vectorization.

The quality-based classifier is constructed using the *Keras* library for deep learning computations, with the *Tensorflow* backend. The result of the experiments are shown in Table 5.

Table 5: *Performance measures of the experiments*

Exp. number	Precision	Recall	Accuracy
Experiment 1.1	65.5%	65.0%	65.0%
Experiment 1.2	73.3%	69.2%	69.3%
Experiment 1.3	73.1%	67.8%	67.8%
Experiment 2.1	75.0%	74.0%	74.0%

Ponzanelli et al. [22] used various metrics for classification. They established three categories of metrics such as *common-metrics, readability-metrics*, and *popularity-metrics*. Although, the last category is related to the questioners, not the questions themselves. They obtained a precision between 61.2% and 66.3% using the decision tree algorithm combined with the common- and the readability-metrics. Using the quality function, they obtained a 73.3% precision regarding the low-quality question based on common metrics. Our results (75% precision) are better than Ponzanellis' in the case of using only the information encoded in the input text. When authors used popularity metrics, they obtained exquisite precision on the right tail of the dataset (up to 90.1%). The main conclusion is that popularity metrics can be considered a better feature in deciding the quality of a given question.

Based on the research the following thesis can be established:

- Neural networks based on the recurrent neural models are adequate tools for classifying the quality of the questions published on Stack Overflow, leaning solely on the textual information encoded in the questions.

3.2 Experiments related to predicting the likelihood of closing

The dataset was prepared differently to classify the likelihood of closing and determine the closing reasons. The dump used for the classification was the same in the previous case, but the filtering and preparation processes differed. Both open and closed questions posted before June 2013 – the introduction of the currently available closing policy and reasons – were filtered. Questions closed due to duplication were also excluded. As already mentioned, neither the online database nor the dump contains the deleted posts, which are only accessible through moderation tools. The deleted data naturally have an impact on accuracy, as the application of deep learning requires a significant amount of data to achieve the possible accuracy that the model provides. The dataset applied for the classification is heavily unbalanced, containing 98,242 off-topic, 46,389 unclear, 46,026 too broad, and 16,383 opinion-based questions. The final dataset consisted of 207,040 closed questions, and – as usual in machine learning – we balanced our dataset by randomly sampling the same amount of open questions for the binary *(open vs. closed)* classification. Next to the likelihood of closing, the reason for the closing was also predicted in this research. For this purpose, the downsampling strategy was applied for balancing the dataset. The size of the classes is based on the smallest class, i.e., the opinion-based class with 16,383 questions.

After the preprocessing phase, the embedding process was executed. In this experiment, we did not use an external library to perform the embedding, but the *Embedding* layer that is part of the neural network, was responsible for the actual vectorization given the dictionary size and dimension size, meaning that the actual embedding is learned parallel with the other model parameters.

The binary classification's objective is to decide between the open versus closed questions, i.e., questions that meet the community rules versus questions that violate these rules. The second series is a five-class classification predicting the different closing reasons *(off-topic, unclear, too broad, opinion-based)* versus the open state. In both experiments, three different recurrent neural network (RNN) models denoted as UNI, BID, and COMP *(composite)* were defined and used.

The input data was split into train and test sets applying the *stratified k-fold cross-validation strategy* with k = 30 yielding 30 different train and test sets. We have selected k = 30 instead of the typical 10 to obtain a more reliable statistical analysis. In each training process and corresponding evaluation, a distinct arbitrary random seed was chosen to ensure both the stochasticity allowing for the statistical analysis of the results and the reproducibility of the experiments.

Table 6: *Average performance measures (and variances) for the binary classification experiment*

Metric	**UNI**	**BID**	**COMP**
Micro precision	71.87 (0.06)%	71.00 (0.04)%	70.84 (0.08)%
Micro recall	71.87 (0.06)%	71.00 (0.04)%	70.84 (0.08)%
Micro F1	*71.87 (0.06)%*	*71.00 (0.04)%*	*70.84 (0.08)%*
Macro precision	73.78 (0.01)%	73.33 (0.01)%	73.66 (0.01)%
Macro recall	71.87 (0.06)%	71.00 (0.04)%	71.00 (0.08)%
Macro F1	*72.81 (0.02)%*	*72.14 (0.02)%*	*72.22 (0.02)%*
Accuracy	**71.87 (0.06)%**	**71.00 (0.04)%**	**70.84 (0.08)%**

The evaluation metrics of the binary classification experiment are shown in Table 6. The Table presents the averages over the results obtained for the test sets in the *30-fold calculations*. The values in parentheses are the corresponding variances also given as percentages. As simply averaging F1 measures does not hold any information, these values in Table 6 were calculated from the average precision and recall numbers. Italic letters indicate this *a posteriori* nature.

For comparison purposes, the last row of Table 6 also lists the accuracies of the models. As only the pre-submission textual information of posts was used as input, and the classification of the questions was performed directly into closed versus open groups, the number of related studies in the literature is scarce.

Our study presented in the previous subsection that focused on quality classification also used textual information exclusively and obtained an accuracy of 74% for classifying SO posts into good versus weak categories using a deep neural network approach. The definition of the category labels as well as the filtering approach for input creation were adopted from the study of Ponzanelli *et al.* [22] and involved the use of post-submission information during the learning phase, including score and the existence of an accepted answer. Correa and Sureka [23], similar to our work, predicted question closing based on various predictive features including post-submission information and obtained an accuracy of 70.3%. The obsolete *(pre-June 2013)* closing policy was considered in their experiments, and questions from their collection, either closed or open, are not included in the dataset used in this study. In our experiments, in contrast to post-submission or user-related information, we considered only textual features irrespective of metrics such as the experience level of the user or question popularity.

Table 7: *Average performance measures (and variances) for the five-class classification experiment*

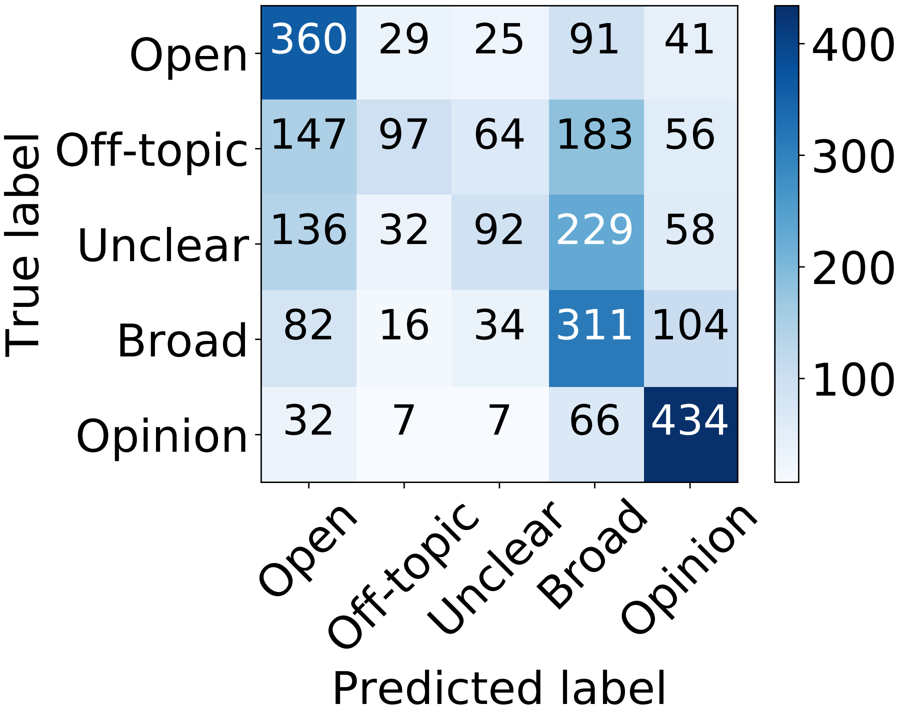
Metric	**UNI**	**BID**	**COMP**
Micro precision	48.55 (0.02)%	47.88 (0.04)%	47.38 (0.03)%
Micro recall	48.55 (0.02)%	47.88 (0.04)%	47.38 (0.03)%
Micro F1	*48.55(0.02)%*	*47.88 (0.04)%*	*47.38 (0.03)%*
Macro precision	50.42 (0.03)%	50.86 (0.02)%	49.04 (0.03)%
Macro recall	48.55 (0.02)%	47.88 (0.04)%	47.38 (0.03)%
Macro F1	*49.47 (0.02)%*	*49.32 (0.03)%*	*48.20 (0.03)%*
Accuracy	**48.55 (0.02)%**	**47.88 (0.04)%**	**47.39 (0.03)%**

The second series of experiments addressed the central question of why exactly a given post will be closed. To this end, a five-class classifier using the same pre-submission textual information as the binary model above is designed with the labels *off-topic*, *unclear*, *too-broad*, *opinion-based*, and *open*. The performance measures on the test set are presented in Table 7. Again, F1 measures written in italics are obtained from the average precision and recall values.

As can be seen, the five-class classifier's performance is far superior to random guessing. There is, however, room for further improvement. One limiting factor to achieving higher accuracy is in distinguishing particular closing reasons. This issue is best represented by an example confusion matrix measured on one of the test sets from the 30-fold run and shown in Figure 4, which displays that the labels *off-topic* and *unclear* are often misclassified.

Although the SO Help Center gives a brief description of the closing reasons, distinguishing them in specific cases can be challenging for both the machine and the human. This assumption is well supported by a study published on the SO Meta site:[1] According

[1] https://meta.stackoverflow.com/questions/390083



	Open	Off-topic	Unclear	Broad	Opinion
Open	360	29	25	91	41
Off-topic	147	97	64	183	56
Unclear	136	32	92	229	58
Broad	82	16	34	311	104
Opinion	32	7	7	66	434

Figure 4: *Confusion matrix for the five-class classification*

to the study, there are many closings without a proper consensus amongst voters, and the efficacy of the voting system is questioned.

The other limiting factor to higher accuracy is the unbalanced dataset of the closing reasons. In contrast to the binary experiment, where all closed questions were considered together, the cardinality of the separate sets in the five-class case is significantly different. In order to improve our model, more data and a better balancing strategy are necessary. One possible way to achieve this is by involving deleted questions in the training set. According to our hypothesis, these posts contain essential characteristics highlighting the features of closed questions. These, however, are not available in the public SO data dumps.

Interestingly, the best results were obtained in our binary and five-class classification experiments with the simplest RNN model, i.e., the unidirectional GRU classifier denoted by UNI.

There is only one study in the literature that directly attempts to predict closure reasons based on textual information [24]. This research was conducted in parallel with our work, and its results became known to us after the publication of our paper. They obtained the best results on balanced data by oversampling the minority class with their neural network-based models. Their result was an average precision of 47%. In our experiments, 48.6% precision was achieved by the UNI model, and both the BID and the COMP model produced slightly better results than Roys' models. Although a direct comparison of the results is not possible due to different datasets, a side-by-side comparison of the results, including the binary and multiclass results, gives an overview in Figure 5.

Based on the research, the following theses can be established:

- Neural networks based on the recurrent neural models are adequate tools for predicting the likelihood of the closures of the questions published on Stack Overflow, leaning solely on the textual information encoded in the questions.
- Neural networks based on the recurrent neural models are proper tools for predicting the possible reasons for the closures of the questions published on Stack Overflow, leaning solely on the textual information encoded in the questions.
- In predicting the reasons for the closure, neural networks deliver more significant errors in cases where the human decision also shows a more uncertain pattern.

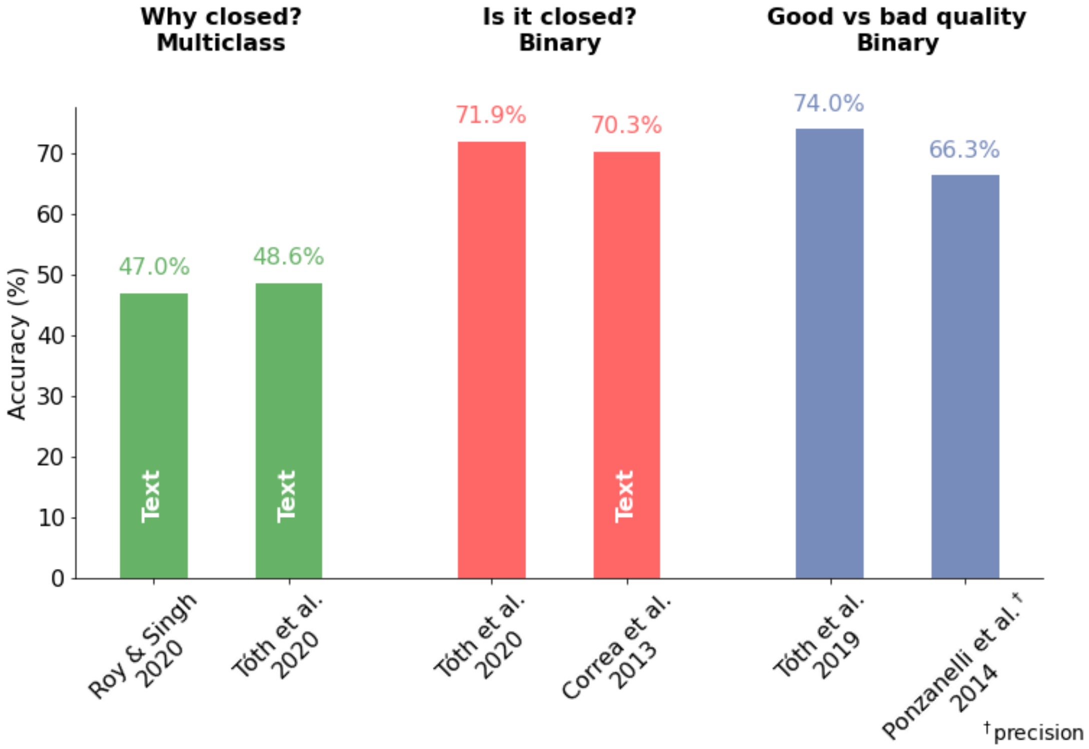


Figure 5: *Comparison of the results with selected literature values*

4 Contributions of the thesis

In the chapter titled **Natural Language Processing and Artificial Intelligence Methods in Requirement Analysis**, my contributions are related to the collection and preparation of the dataset, implementation of the scripts, execution, and evaluation of the experiments. A detailed discussion can be found in Chapter 2.

I/1. The author has developed preprocessing methods and a vectorization process applying the `tf-idf` representation form.

I/2. The author has implemented scripts responsible for executing the classification experiments using various machine learning models implemented in the `scikit-learn` library.

I/3. The author has implemented a simple neural network applied in the classification experiments based on the Stack Overflow samples.

I/4. The author executed the experiments, compared the results of the classifiers, and identified the best classifiers.

I/5. Based on the investigation of the semantics of the linguistic expressions, the author established a solid definition of semantic space and semantic networks, respectively.

I/6. The author has implemented preprocessing steps to extract posts from the Stack Overflow database and separate them into proper sentences cleaned from the auxiliary characters and noise.

I/7. The author has implemented a set of regular expressions based on the lexico-syntactic patterns representing the hyperonym-hyponym relationships found in the literature.

I/8. The author has developed a phrase structure grammar and an automatization to recognize noun phrases in the text. To the best of our knowledge, the grammar provided by the author is the most general formalized solution available in the literature.

I/9. The author has developed a simplified automatization for recognizing the noun phrases considering only a small set of the lexico-syntactic patterns.

I/10. The author has built a semantic network based on the hyperonym-hyponym relationships, representing the semantic field of the software development community based on the Stack Overflow post utilizing the lexico-syntactic patterns.

I/11. The author has investigated the structure of the resulting network and described its structure.

I/12. The author compared the smaller network resulting from the mining process with the semantic network representing the common knowledge provided by WordNet.

In the chapter titled **Investigating Developers Interactions Using Natural Language Processing And Deep Learning**, my contributions are related to preprocessing the dataset used in the experiments, creating deep learning models, executing the prediction, and evaluating the results. A detailed discussion can be found in Chapter 3.

II/1. The author has developed a GRU-based deep learning model to classify the questions based on their quality posted to Stack Overflow, considering only the textual elements. The definition of quality was taken from the literature for comparison purposes.

II/2. The author executed the classification by applying different amounts of samples and vectorization processes.

Table 8: *Correspondence between the thesis points and my publications.*

Publication	Thesis point																	
	I/1	I/2	I/3	I/4	I/5	I/6	I/7	I/8	I/9	I/10	I/11	I/12	II/1	II/2	II/3	II/4	II/5	II/6
[1]			●	●														
[2]		●		●														
[3]													●	●	●			
[4]																●	●	●
[5]					●	●	●	●		●								
[6]	●																	
[7]						●			●	●	●	●						

II/3. The author compared the results with each other and the results of the other researchers. When all inputs were used in the classification process, the model's performance outperformed the performance of the classifiers used by others, demonstrating that deep learning solutions can provide better results if there is enough input available.

II/4. The author has developed three distinct GRU-based deep learning models to classify the likelihood of closing questions posted to the Stack Overflow, considering only the textual information available during the assembling of that question.

II/5. The author executed the classifications and compared the results with the other results available in the literature. The classifier provided good performances outperforming the other results, which can also be used in practice to evaluate the possibility of the question's closure before posting it to Stack Overflow.

II/6. The author modified the models to perform multi-class classification. In this case, the classifiers predict the possible closing reasons. To the best of our knowledge, this was the first classifier published to predict the closing reasons. However, a parallel study was in progress and published lately for the same purpose, but the classifiers made by the author have a slightly better performance. The classifiers provide a good result and are applicable in practice.

Table 8 summarizes the relation between the thesis points and the corresponding publications.

The author's publications on the subjects of the thesis

Journal publications

[1] **László Tóth** and László Vidács Comparative Study of The Performance of Various Classifiers in Labeling Non-Functional Requirements. *Information Technology and Control*, 48(3), 432-445, 2019.

Papers in conference proceedings

[2] **László Tóth** and László Vidács Study of various classifiers for identification and classification of non-functional requirements. In *Computational Science and Its Applications – ICCSA 2018*, Springer, 492-503, 2018.

[3] **László Tóth**, Balázs Nagy, Dávid Janthó, László Vidács, Tibor Gyimóthy Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach. In *Proceedings of the 14th International Conference on Software Technologies*, Institute for Systems and Technologies of Information, Control and Communication, 631-639, 2019.

[4] **László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács** Why Will My Question Be Closed? NLP-Based Pre-Submission Predictions of Question Closing Reasons on Stack Overflow. In *2020 ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results - ICSE-NIER* Association for Computing Machinery (ACM), 45-48, 2020.

[5] **László Tóth** and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In *The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021*, IARIA, 1-6, 2021.

Further related publications

[6] **László Tóth** Preliminary Concepts for Requirements Mining and Classification using Hidden Markov Model. In *11th Conference of PhD Students in Computer Science*, 110-113, 2018.

[7] **László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács** Mining Hypernyms Semantic Relations from Stack Overflow. In *2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops - KG4SE* Association for Computing Machinery (ACM), 360-366, 2020.

Other References

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

[9] Donald Firesmith. Common Requirements Problems, their Negative Consequences, and The Industry Best Practices to Help Solve Them. *Journal of Object Technology*, 6 (1):17–33, 2007. doi 10.5381/jot.2007.6.1.c2.

[10] Azham Hussain, Emmanuel Mkpojiogu, and Fazillah Kamal. The Role of Requirements in The Success or Failure of Software Projects. *EJ Econjournals*, 6(7S):6–7, 2016.

[11] Steve Bundred. Solutions to Silos: Joining up Knowledge. *Public Money & Management*, 26(2):125–130, 2016. ISSN 0954-0962 doi 10.1111/j.1467-9302.2006.00511.x.

[12] Tshidi Mohapeloa. Effects of Silo Mentality on Corporate ITC's Business Model. *Proceedings of the International Conference on Business Excellence*, 11(1):1009–1019, 2017. doi 10.1515/picbe-2017-0105.

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

[14] Edna Dias Canedo and Bruno Cordeiro Mendes. Software requirements classification using machine learning algorithms. *Entropy*, 22(9), 2020. ISSN 1099-4300 doi 10.3390/e22091057.

[15] B Caglayan, E Kocaguneli, J Krall, Fayola Peters, and Burak Turhan. The PROMISE Repository of Empirical Software Engineering Data, 2012.

[16] Agustin Casamayor, Daniela Godoy, and Marcelo Campo. Identification of Non-Functional Requirements in Textual Specifications: A Semi-supervised Learning Approach. *Information and Software Technology*, 52(4):436–445, 2010. doi 10.1016/J.INFSOF.2009.10.010.

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

[18] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 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*. European Language Resources Association {(ELRA)}, 2016.

[19] Princeton University. About WordNet, 2010. URL `https://wordnet.princeton.edu/`.

[20] I. Srba and M. Bielikova. Why is Stack Overflow Failing? Preserving Sustainability in Community Question Answering. *IEEE Software*. doi 10.1109/MS.2016.34.

[21] Aleksi Aaltonen and Sunil Wattal. Rejecting and Retaining New Contributors in Open Knowledge Collaboration: A Natural Experiment in Stack Overflow Q&A Service. In *ECIS 2020 Research Papers*, volume 183, page 16, 2020.

[22] Luca Ponzanelli, Andrea Mocci, Alberto Bacchelli, and Michele Lanza. Understanding and Classifying the Quality of Technical Forum Questions. In *14th International Conference on Quality Software*, pages 343–352. IEEE, oct 2014. ISBN: 978-1-4799-7198-5, doi: 10.1109/QSIC.2014.27.

[23] Denzil Correa and Ashish Sureka. 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, pages 201–212, New York, NY, USA, 2013. ACM. ISBN: 978-1-4503-2084-9, doi: 10.1145/2512938.2512954.

[24] Pradeep Kumar Roy and Jyoti Prakash Singh. Predicting Closed Questions on Community Question Answering Sites Using Convolutional Neural Network. *Neural Computing and Applications*, 32(14):10555–10572, Jul 2020. ISSN 1433-3058 doi 10.1007/s00521-019-04592-0.

5 Összefoglalás

Az értekezés két kutatási vonalat ismertet. A természetes nyelvfeldolgozó eljárások és a mesterséges intelligencia módszereinek alkalmazása a követelményanalízisben című fejezetben két olyan kutatási vonal eredményeinek ismertetésére került sor, amelyek célja a követelményelemzés folyamatának támogatása annak érdekében, hogy a természetes nyelven adott követelmények feldolgozása pontosabb és teljesebb legyen. Kutatásunk egyik részében azt a célt tűztük ki, hogy a sokszor elnagyolt, nem megfelelően hangsúlyozott nem funkcionális követelmények felismerésének és osztályozásának folyamatára automatizálható eljárásokat dolgozzunk ki. Ennek keretében különböző gépi tanuló eljárások alkalmazhatóságát vizsgáltuk és megállapítottuk, hogy azok alkalmasak a nem funkcionális követelményeket tartalmazó, természetes nyelven adott mondatok felismerésére és a nem funkcionális követelmények osztályozására.

A követelmények megfogalmazása általában az adott üzleti környezet nyelvhasználati sajátosságának figyelembevételével történik. Az egyes szemantikai környezetben azonban gyakran ugyanazok a nyelvi kifejezések mást jelentenek, vagy olyan hallgatólagos tudásra is utalhatnak, amelyekkel a fejlesztők nem rendelkeznek. Kutatásunkban megvizsgáltuk a kommunikációs folyamatban akadályt képező, a nyelvhasználat szemantikai viszonyait érintő tényezőket, pontos definíciót adtunk a szemantikus tér és szemantikus hálózat fogalmáról, illetőleg a szoftverfejlesztés szemantikai környezetét demonstráló szemantikus hálózatot építettünk fel a Stack Overflow posztok felhasználásával. A kapott hálózat felépítését megvizsgáltuk, illetőleg kimutattuk, hogy az abban található terminusok jelentése eltér a köznapi használattól. Ennek a jelenségnek kimutatására a köznapi jelentést demonstráló WordNet hálózatot használtuk.

A fejlesztői interakciók vizsgálata természetes nyelvfeldolgozó módszerek és a mélytanulás alkalmazásával című fejezetben a szoftverfejlesztők Stack Overflow-n zajló kommunikációs folyamatait vizsgáltuk, konkrétabban a portálra feltett kérdések minőségét, azok lezárási lehetőségeit, illetőleg annak lehetséges okait. Kutatásunkban arra helyeztük a hangsúlyt, hogy a kérdések összeállításának idejében ismert adatokból fel tudjuk becsülni az adott kérdés minőségét, illetőleg azt, hogy az adott kérdést várhatóan lezárják-e és ha igen, akkor milyen indokkal. A kutatás két irányban zajlott. Először a kérdések minőségének becslését végeztük el mélytanulás segítségével a szakirodalomban ismert minőségi fogalmat felhasználva. A modellünk eredményét összehasonlítottuk az ismert eredményekkel, amelyekből az derült ki, hogy az általunk használt mélytanulásra épülő modell pontosabb becslést ad a konkrurrens modelleknél.

A minőség azonban nem ad pontos előrejelzést, ha a kérdés lehetséges lezárását tekintjük, ugyanis magas minőségű kérdéseket is lezárhatnak például azért, mert nem témába illők. Kutatásunk során ezért feljlesztettünk mind a vektorizálás folyamatán, mind az alkalmazott neurális hálók architektúráján és a kapott modellekkel a lezárás közvetlen lehetőségét vizsgáltuk. A bináris osztályozás esetén a szakirodalomban fellelhető osztályozóknál jobb eredményeket sikerült elérni. A többosztályos esetben, mikor a kérdések lezárásának okát is vizsgáltuk az eredményeink mérsékeltebbek, azonban ebben az esetben is jól alkalmazhatók a gyakorlatban. A kísérletek időpontjában hasonló kísérleteket nem publikáltak, munkánkkal párhuzamosan zajlott egy kutatás, amely hasonlóan többosztályos osztályozó alkalmazására épült, azonban a mi eredményeinknél a szerzőknek mérsékeltebb eredményt sikerült elérniük.

Nyilatkozat

Tóth László *Natural Language Processing and Artificial Intelligence Methods in Software Engineering* című PhD disszertációjában a következő eredményekben **Tóth László** hozzájárulása volt a meghatározó:

Az **első tézisponthoz** (*Natural Language Processing and Artificial Intelligence Methods in Requirements Analysis*) tartozó eredmények:

1. A tf-idf reprezentációra épülő előfeldolgozó, valamint vektorizáló eljárások kidolgozása és implementációja.
    - László Tóth Preliminary Concepts for Requirements Mininf and Classification using Hidden Markov Model. In 11th Conference of PhD Students in Computer Science, 110-113, 2018
2. A scikit-learn library-ben implementált gépi tanuló modellekre épülő osztályozó kísérleteket futtató scriptek elkészítése.
    - László Tóth and László Vidács Study of various classifiers for identification and classification of non-functional requirements. In Computational Science and Its Application – ICCSA 2018, Springer, 492-503, 2018
3. A Stack Overflow mintákat használó kísérletek végrehajtásához egyszerű neurális hálózat elkészítése.
    - László Tóth and László Vidács Comparative Study of The Performance of Various Classifiers in Labeling Non-Functional Requirements. Information Technology and Control, 48(3), 432-445, 2019.
4. Osztályozási kísérletek elvégzése, eredmények összehasonlítása, a legjobb eredményeket adó modellek azonosítása.
    - László Tóth and László Vidács Study of various classifiers for identification and classification of non-functional requirements. In Computational Science and Its Application – ICCSA 2018, Springer, 492-503, 2018
    - László Tóth and László Vidács Comparative Study of The Performance of Various Classifiers in Labeling Non-Functional Requirements. Information Technology and Control, 48(3), 432-445, 2019.
5. A szemantikus tér és a szemantikus hálózat fogalmának pontos meghatározása.
    - László Tóth and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021, IARIA, 1-6, 2021

6. Előfeldolgozó eljárások készítése a Stack Overflow posztok kivonatolására, amely eljárások célja az adatok zajmentesítése, a felesleges elemek eltávolítása és az input mondatokra tagolása.
    - László Tóth and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021, IARIA, 1-6, 2021
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Mining Hypernyms Semantic Relations from Stack Overflow. In 2020 IEEE/ACM 42n International Conference on Software Engineering Workshops – KG4SE Association for Computing Machinery (ACM), 360-366, 2020
7. Az irodalomban fellelhető lexiko-szintaktikus minták implementálása reguláris kifejezések segítségével.
    - László Tóth and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021, IARIA, 1-6, 2021
8. Kifejezés struktúrájú nyelvtan, valamint a kapcsolódó determinisztikus automata definiálása és implementálása főnévi kifejezések felismeréséhez.
    - László Tóth and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021, IARIA, 1-6, 2021
9. Egyszerűsített automata implementálása főnévi kifejezése felismeréséhez szűkebb halmazból vett lexiko-szintaktikus minták alkalmazása mellett.
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Mining Hypernyms Semantic Relations from Stack Overflow. In 2020 IEEE/ACM 42n International Conference on Software Engineering Workshops – KG4SE Association for Computing Machinery (ACM), 360-366, 2020
10. A Stack Overflow posztok alapján a szoftverfejlesztés szemantikus környezetének reprezentációját megvalósító, hiperním-hiponím kapcsolatokra épülő szemantikus hálózat készítése.
    - László Tóth and László Vidács Analyzing Hyperonyms of Stack Overflow Posts. In The Seventh International Conference on Fundamentals and Advances in Software Systems Integration – FASSI 2021, IARIA, 1-6, 2021
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Mining Hypernyms Semantic Relations from Stack Overflow. In 2020 IEEE/ACM 42n International Conference on Software Engineering Workshops – KG4SE Association for Computing Machinery (ACM), 360-366, 2020
11. Az elkészült szemantikus hálózat felépítésének vizsgálata és dokumentálása.
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Mining Hypernyms Semantic Relations from Stack Overflow. In 2020 IEEE/ACM 42n International Conference on Software Engineering Workshops – KG4SE Association for Computing Machinery (ACM), 360-366, 2020
12. A szemantikus hálózat és a WordNet által reprezentált szemantikus hálózat összehasonlítása, a kontextus szerepének kimutatása.

- László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Mining Hypernyms Semantic Relations from Stack Overflow. In 2020 IEEE/ACM 42n International Conference on Software Engineering Workshops – KG4SE Association for Computing Machinery (ACM), 360-366, 2020

A **második tézisponthoz** (*Investigating Developers Interactions Using Natural Language Processing And Deep Learning*) tartozó eredmények:

1. GRU alapú mélytanuló modell készítése a Stack Overflow portálon publikált kérdések minőségének szöveges információn alapuló osztályozásához. A minőség definícióját a szakirodalom alapján vette a jelölt.
    - László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, Tibor Gyimóthy Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach. In Proceedings of the 14th International Conference on Software Technologies, Institute for Systems and Technologies of Information, Control and Communication, 631-639, 2019
2. Minőség alapú osztályozási kísérletek elvégzése különböző számosságú mintával, valamint eltérő vektorizációs eljárásokat alkalmazva.
    - László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, Tibor Gyimóthy Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach. In Proceedings of the 14th International Conference on Software Technologies, Institute for Systems and Technologies of Information, Control and Communication, 631-639, 2019
3. Az 2. pontban említett kísérletek eredményeinek egymással, valamint a szakirodalomban publikált eredményekkel történő összevetése, a mélytanuló modellek jobb teljesítményének kimutatása.
    - László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, Tibor Gyimóthy Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach. In Proceedings of the 14th International Conference on Software Technologies, Institute for Systems and Technologies of Information, Control and Communication, 631-639, 2019
4. Három GRU alapú modell készítése a Stack Overflow-n publikált kérdések lezárásának előrejelzésére, kizárólag a szöveges információkra támaszkodva.
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Why Will My Question Be Closed? NLP-Based Pre-Submission Predictions of Question Closing Reasons on Stack Overflow. In 2020 ACM/IEEE 42nd International COnference on Software Engineering: New Ideas and Emerging Results – ICSE – NIER Association for Computing Machinery (ACM), 45-48, 2020

5. Kísérletek elvégzése, eredmények összehasonlítása egymással és az irodalomban publikált eredményekkel.
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Why Will My Question Be Closed? NLP-Based Pre-Submission Predictions of Question Closing Reasons on Stack Overflow. In 2020 ACM/IEEE 42nd International COnference on Software Engineering: New Ideas and Emerging Results – ICSE – NIER Association for Computing Machinery (ACM), 45-48, 2020
6. Modellek módosítása többosztályos kísérletekhez, ahol a lehetséges lezárások okának előrejelzése is megtörténik, továbbra is kizárólagosan szöveges információkra támaszkodva, valamint a kísérletek végrehajtása.
    - László Tóth, Balázs Nagy, Tibor Gyimóthy, László Vidács Why Will My Question Be Closed? NLP-Based Pre-Submission Predictions of Question Closing Reasons on Stack Overflow. In 2020 ACM/IEEE 42nd International COnference on Software Engineering: New Ideas and Emerging Results – ICSE – NIER Association for Computing Machinery (ACM), 45-48, 2020

Ezek az eredmények **Tóth László** PhD disszertációján kívül más tudományos fokozat megszerzésére nem használhatók fel.

Szeged, 2022.03.21

Tóth László
jelölt

Dr. Gyimóthy Tibor
témavezető

Dr. Vidács László
témavezető

Az Informatika Doktori Iskola vezetője kijelenti, hogy jelen nyilatkozatot minden társszerzőhöz eljuttatta, és azzal szemben egyetlen társszerző sem emelt kifogást.

Szeged, 2022.03.24.

SZEGEDI TUDOMÁNYEGYETEM
INFORMATIKA DOKTORI ISKOLA
SZEGED

Dr. Jelasity Márk
Informatikai Doktori Iskola vezetője