

## EVOLUTION OF SOFTWARE QUALITY MODELS FROM AN NLP PERSPECTIVE

D. LUPȘA, S. MOTOGNA, AND I. CIUCIU

**ABSTRACT.** The research question addressed in this study is to analyze the evolution of software quality models, and to investigate the similarities between the most known quality models. The models are decomposed according to their inherent structure, then are analyzed based on their subcharacteristics, at several levels. The focus of the study is on the lexical analysis of software quality models based on natural language processing, with the purpose to highlight the similarities in software quality models. In the end, the current software quality model ISO25010 is compared with all previous ones.

### 1. INTRODUCTION

As software systems become every day, more and more complex and spread in different application areas, the interest for software quality (SQ) has increased significantly.

From a business perspective, quality aspects of software applications, such as reliability, usability, interoperability can bring significant advantages over the competition. From an industrial perspective, different software quality characteristics, such as security, safety, maintainability, are essential in some types of applications: real-time, safety-critical, respectively health support systems. Even more, there are standardized software quality models for software products (ISO25010[9]) and processes. From the research perspective, the last decade has seen an increased interest of the research community for software quality aspects, with different conferences and journals focusing on SQ in general, but also on specific SQ characteristics such as: reliability, security, safety a.s.o.

A software quality model represents a set of characteristics that completely characterize a software system. They are classified as external factors (such

---

Received by the editors: 1 June 2020.

2010 *Mathematics Subject Classification.* 68N99.

1998 *CR Categories and Descriptors.* D.2.0 [**Software Engineering**]: General – *Standards*; D.2.9 [**Software Engineering**]: Management – *Software Quality Assurance*.

*Key words and phrases.* Software Quality, quality factors, word similarity.

as usability, reliability, efficiency a.s.o.) and internal (including maintainability, testability, reusability a.s.o.). The characteristics are then divided into subcharacteristics, with the intent to cover all aspects of software products.

Software development evolution has a direct impact on software models, with several SQ models proposed, and lately, standardized.

The aim of this article is to study the evolution of software quality models using an NLP approach. Our aim is to be able to explain the similarities and variations between existing SQ models by looking at both their syntactical and semantical structure. More specifically, our approach decomposes a model according to its inherent structure (i.e. characteristics and subcharacteristics) and tries to interpret (analyze) it at lexical level, by applying various NLP metrics. This study is a continuation of our previous work [15], in terms of the quantitative analysis that is employed and also regarding the software quality models under investigation.

The rest of the paper is structured as follows: we begin with a short presentation of the SQ models under review, followed by the Natural Language Processing (NLP) measurements used in our investigation. The next section discusses our finding related to models evolution, and presents the similarity study performed for two models: McCall and ISO 25010. We end with some concluding remarks and future ideas.

## 2. SQ MODELS

The McCall model[12] introduced in 1977 is consider the first software quality model, defining 11 factors characterizing software products, that are further decomposed in criteria, as an initiative from aeronautical industry. Several notable models followed, proposing new factors or modifying the ones from McCall model. A synthesis of the most important SQ models, including their component factors, is presented in Table 1.

Our study will include the following SQ models: McCall, Boehm, ISO 9126 and ISO 25010. The decision to exclude FURPS model is based on the fact that it was initially developed as an "in-the-house" model at Hewlett Packard, and although it is widely used in industry the definition of characteristics and their evaluation varies from company to company.

Looking at the factors, we can notice that some of them, like maintainability and reliability, are quite "stable", namely have been included in all proposed quality models, while other factors appeared only in a subset of these models. This can be explained by the dynamics of software domain, which changes very fast. For example security appeared in the last SQ model, due to the spread of web and mobile applications. A deeper look also identifies similarities or differences between subcharacteristics.

TABLE 1. Software quality models [15]

SQ Model	Year	Factors
McCall [12]	1977	Correctness, Reliability, Efficiency, Integrity, Usability, Maintainability, Testability, Flexibility, Portability, Reusab., Interoperability
Boehm [2]	1977	Portability, As in Utility - containing Reliability, Efficiency, Human Engineering, Maintainability - including Testability, Understandab., Flexibility
FURPS [8]	1987	Functionality(F), Usability (U), Reliability (R), Performance (P) and Support (S)
ISO 9126 [1]	2001	Functionality, Reliability, Usability, Efficiency, Maintainability, Portability
ISO 25010 [9]	2011	Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, Portability

Analyzing the SQ models, we can conclude the following:

- The domain of software quality uses several concepts and terms with different definitions and meaning;
- The domain of software quality is dynamic: software development rapid growth directly influences SQ domain.

**2.1. State of the art.** Regarding the evolution of SQ models, a study taking into account the hierarchy of characteristics-subcharacteristics, relations of subordination between them and special metrics is performed in [6] in the context of ISO 25010, where the need of a more detailed semantic analysis for the disambiguation of quality model components is recognized. The same study was applied in the context of “green software” in [7], and managed to predict the complexity for the next software quality model and to propose variants of green software attributes to be included in the model.

It is agreed within the software quality community that in order to apply semantic analysis on SQ models, it is necessary to have a commonly agreed-upon model, i.e. an ontology. However, such model is currently not completely available. Several contributions address the topic of constructing a common set of concepts that are incorporated in quality models. A study [11] that employs extraction of a condensed model for software product quality attributes is using the frequency and association of terminology and concepts that appear in different sources such as models, standards and related documents to software quality. Based on this, an ontology involving semantics of quality attributes is built. This ontology was the starting point for further studies targeting a common understanding and agreed models to be used by software

engineers, researchers, stakeholders and practitioners. A three-dimensional ontology model was proposed in [3], SQuAP-Ont (<https://w3id.org/squap/>), based on current standards in place for software quality, processes and architecture. For software quality, SQuAP is using the ISO25010 standard, and can be easily extendable. Kara et al. [10] proposed a generic software quality model based on existing software standards that can be instantiated using an algorithm based on fuzzy logic.

The novelty of our study is given by the fact that the analysis is considering all significant SQ models, in an attempt to identify similarities and differences from a NLP perspective, which will help unify the terminology used in this domain.

### 3. INTRODUCING NLP MEASURES

As discussed in [15], several NLP approaches can be considered for software quality domain, computing similarities between characteristics and subcharacteristics. These computations can be classified in:

- set-based similarities [5]: they are using mostly the number of common words. No semantic information is considered. The intuition behind them is that the more common words two texts share, the more similar they are;
- lexical similarities: they can be computed based on existing ontologies (such as WordNet [4]) or can be computed based on the word co-occurrences in texts, as in the case of word embedding techniques.

The experiments from [15] showed that set-based measures, based on keywords, give the most accurate results for set-based similarity. That is the argument why the rest of the study will consider characteristics and subcharacteristics as keywords and use only them to compute similarity between SQ models.

**3.1. Set-based measures.** Overlap, Jaccard and Dice are set-based similarity measures that can be used.

Overlap coefficient is computed based on the number of common terms and uses a normalization factor, but it considers two sets a full match, having maximum similarity value, which is 1, if one is a subset of the other.

Jaccard similarity is computed as the number of shared (common) keywords over the number of all unique keywords (union) in two sets, and can be defined in terms of what is the same (common) and what is modified (added or removed) when having a SQ model SQM1 and creating a second model SQM2.

Dice's coefficient is defined as twice the number of common terms in the compared sets divided by the total number of terms in both sets, which is also:

$$(1) \quad \text{Dice}(SQM1, SQM2) = \frac{2 \times \text{common}(SQM1, SQM2)}{2 \times \text{common}(SQM1, SQM2) + \text{added}(SQM1, SQM2) + \text{removed}(SQM1, SQM2)}$$

Both Dice and Jaccard depend on the number of items that are modified (added or removed) and they decrease when this number increases, but Dice has higher values than Jaccard. It is also proven that Jaccard and Dice are monotonic in one another, meaning that it not exists 3 sets A, B and C such that:  $\text{Jaccard}(A; B) > \text{Jaccard}(A; C)$  and  $\text{Dice}(A; B) < \text{Dice}(A; C)$ .

Dice and Jaccard give a similar kind of information and we decided to use one of them, namely Dice.

**3.2. Lexical information.** One of the things we are aiming at is to identify renaming of the same concept in SQ models, for example, to identify words like *maintainability* and *changeability* as being synonyms (that is lexically similar).

There are a lot of measures to determine the similarity between words. Some of them use existing ontologies (such as WordNet) to compute the semantic distance between words. Others are based on the word co-occurrences in texts, such as word embedding techniques.

Word embedding techniques were introduced around 2013 by a team led by Mikolov that proposed the word2vec algorithm [13], [14]. It was followed by the apparition of other unsupervised learning algorithms for obtaining vector representations for words, such as GloVe [16]. These methods obtain good results, but they rely on having a very large corpus. But *How large the corpus should be?* in order to cover the semantic information that is present in human-build ontologies (such as WordNet). For example, we would expect to get a high similarity value between words like *change* and *modify*, but let us look to the data. By using one of Standford GloVe word embeddings vectors (Wikipedia 2014 + Gigaword 5) the similarity between *change* and *modify* is 0.461, while the most 5 similar words with modify and their similarity are: (*modifying*, 0.791), (*amend*, 0.751), (*revise*, 0.728), (*alter*, 0.717), (*rework*, 0.672) and the most 5 similar words with change are: (*changes*, 0.904), (*changing*, 0.809), (*this*, 0.787), (*shift*, 0.787), (*move*, 0.785).

The amount of data corresponding to this study is limited, so we decided that such an approach may lead to false results, and as a consequence we decided to use WordNet.

Since we aim at identifying the renaming of the same concept in SQ models and we need information about synonymy between words, and we choose to use WordNet. WordNet has the advantage that is human build ontology (and

it means that we can trust the data), it covers our needs and it is ready-to-use. If two words appear in the same WordNet synset, they are considered synonyms and describe the same concept.

We cannot identify all the synonymy relations that we need directly by searching WordNet, because there are a lot of missing words. But we expect that most of the missing words in WordNet to be derivations of others, which we are going to call base word. For example, we can't find *modifiability* in WordNet, but we can find its base word: *modify* and also can find that *change* and *modify* are synonyms. With this approach, we were able to find in WordNet all the (base) words that are important for us.

So, in this study, in order to find and use lexical information, we first took all the base words for all the names of characteristics and subcharacteristics and then verify if they are synonyms or not.

Another problem we had to face is the use of multi-word expression for characteristics and subcharacteristics names. As they could be considered as containing only key-names for a concept, our choice was to consider as being lexically similar, expressions that have at least one (base) word in common or, at least, lexically similar.

By associating all these key-names that are similar from a lexical perspective to a given key-name, many times there will be more than one association to a name. Consider that we have a number  $na$  of names that are in set  $A$ , and for which there is at least a similar name in a set  $B$ . Denote with  $nb$  all the words from  $B$  that are lexical associations to  $A$ . We have to remark that, many times,  $nb \neq na$ .

When we compare a model SQM1 to SQM2, we consider the number of key-expressions from the SQM1 for which we find a lexically similar key-expression in SQM2. For example, in Table 7, column Boehm vs. 9126, second half of the table, named: use lexical information, we found lexical association for 5 characteristics from Boehm into 9126 and we didn't find any for 4 characteristics of Boehm. So, there are 5 lexical matches and 4 removed ones. On the line *lexical match* there is also a number put in paranthesis: (vs. 6). That number 6 means that for all the 6 characteristics from 9126 there are lexical associations in Boehm and that is nothing left in 9126 that has no association in Boehm. This also means that there is no new characteristic added to Boehm.

We cannot directly use Dice as a metric. We are modifying it as follows: instead of counting the common names, we are going to number all the names from the two sets for which there are lexically similar names in the other set. For example, Table 7, Boehm vs. 916, modified Dice is  $(5 + 6)/(5 + 6 + 0 + 4)$ .

## 4. EVALUATION DATA. COMPARATIVE ANALYSIS OF SQ MODELS

We considered an investigation of transformations between the models. A first aspect of the analysis takes into consideration questions like: how many properties remain the same, how many are added and how many are removed. Another question to be answered is: how do we define the properties of a model? In this approach we are going to look at the characteristics and sub-characteristics names and relations among names from a lexical perspective.

A second aspect of the analysis is the perspective from which we look at the data. We consider the SQ models: McCall, Boehm, ISO 9126, ISO 25010 taken in historical order, but also analyze how close models are relative to the latest model among them, which is ISO 25010. The data has been collected using naming of characteristics and subcharacteristics.

TABLE 2. Naming of the characteristics and subcharacteristics. Evolution

		McCall to Boehm	Boehm to 9126	9126 to 25010
names	common	11	7	18
	added	12	20	21
	removed	23	16	9
	Dice	0.39	0.28	0.55
use lexical information	lexical match	20	9	23
	added	7	15	11
	removed	14	14	4
	modified Dice	0.63	0.42	0.77

**4.1. Quantitative analysis based on all names of characteristics and subcharacteristics.** We can see in Table 2 that, from McCall to Boehm, 35 naming were changed (added or removed), which are a little less than from Boehm to ISO 9126 (36), and only 30 are changed between ISO 9126 and ISO 25010, which seemed to be the most similar software quality models among those considered here. This behaviour of the evolution is even more pregnant if we look at the values obtained by using lexical information.

When compared with ISO 25010, the biggest differences are found for Boehm model. We can also notice that ISO25010 is the model that preserved the most of characteristics (Tables 2 and 3 ) during all models history.

If we look to the common names (words) chosen for characteristics and subcharacteristics in all four models, we see that there are only four names that appear in all of them, and these are: maintainability, portability, reliability, testability, and only two of them remained as characteristics in all models.

TABLE 3. Naming of characteristics and subcharacteristics compared to ISO 25010

		McCall vs. 25010	Boehm vs. 25010	9126 vs. 25010
names	common	10	7	18
	added	29	32	21
	removed	24	16	9
	Dice	0.27	0.23	0.55
use lexical information	lexical match	19	11	23
	added	18	21	11
	removed	15	12	4
	modified Dice	0.55	0.47	0.77

TABLE 4. Common characteristics and subcharacteristics in SQ Models

	McCall	Boehm	9126	25010
maintainability	factor	superfactor	factor	characteristic
portability	factor	factor	factor	characteristic
reliability	factor	factor	factor	characteristic
testability	factor	factor	subfactor	subcharacteristic

#### 4.2. Quantitative analysis based on all names of subcharacteristics.

In this section we consider only subcharacteristics names, as a characteristics can be considered satisfied when all its subcharacteristics are satisfied. An interesting remark: no subcharacteristics appears in all models as subcharacteristics. Only Testability, which is a subcharacteristic of Maintainability in both ISO 9126 and in ISO 25010, appears in McCall and in Boehm models, but as a subcharacteristics.

More than that, only two subcharacteristics appear (by the same name) in three of the four models: accuracy (in McCall, Boehm, ISO 9126) and operability (McCall, ISO 9126, ISO 25010). An interesting fact is that 43 subcharacteristics names (from 65) are used only in one model, and 20 are used in two.

#### 4.3. Quantitative analysis based on all names of characteristics.

The list of characteristics of a SQ model can give an overview of the aspects on which that model focus on. A comparison of what is added, removed or kept along time and how it differs from ISO 25010, in the same style as before, is presented in what follows.

This analysis shows that if we look only at the level of characteristics the SQ models may seem more similar than they really are. As presented in Table 7,

TABLE 5. Naming of the subcharacteristics. Evolution

		McCall to Boehm	Boehm to 9126	9126 to 25010
names	common	6	1	13
	added	8	20	18
	removed	17	13	8
	Dice	0.32	0.06	0.50
use lexical information	lexical match	20	9	18
	added	7	15	10
	removed	14	14	9
	modified Dice	0.63	0.42	0.67

TABLE 6. Naming of the subcharacteristics versus ISO 25010

		McCall vs. 25010	Boehm vs. 25010	9126 vs. 25010
names	common	2	2	13
	added	29	29	18
	removed	21	12	8
	Dice	0.07	0.09	0.50
use lexical information	lexical match	14	8	18
	added	16	19	10
	removed	20	15	9
	modified Dice	0.45	0.37	0.67

TABLE 7. Naming of the characteristics. Evolution

		McCall to Boehm	Boehm to 9126	9126 to 25010
names	common	5	4	4
	added	4	2	4
	removed	6	5	2
	Dice	0.50	0.53	0.57
use lexical information	lexical match	6 (vs. 6)	5 (vs. 6)	6 (vs. 6)
	added	3	0	2
	removed	5	4	0
	modified Dice	0.6	0.73	0.86

the Dice similarity values are very close to each other and, in case of comparing McCall with Boehm, respectively Boehm with ISO 9126 models, much greater than the values from the previous comparisons.

The same increase in similarity for characteristics can be observed when comparing previous SQ models with ISO 25010, as shown in Table 8.

TABLE 8. Naming of the characteristics versus ISO 25010

		McCall vs. 25010	Boehm vs. 25010	9126 vs. 25010
names	common	4	3	4
	added	4	5	4
	removed	7	6	2
	Dice	0.42	0.35	0.57
use lexical information	lexical match	5 (vs. 6)	5 (vs. 6)	6 (vs.6)
	added	2	2	2
	removed	6	4	0
	modified Dice	0.58	0.65	0.86

## 5. COMPARATIVE ANALYSIS OF ISO25010

A conclusion of the evaluations from the last section, when comparing models with ISO 25010, can be summarized in:

The most similar model is ISO 9126, the direct predecessor in terms of evolution, as shown in Tables 3, 6, 8. This is something to be expected, but however if we make the comparison with the modifications between the other consecutive models, we notice that in almost all the cases (the exception is the total number of characteristics and subcharacteristics added, as shown in Table 2) the modifications are fewer and the common elements are the most. Thus, we can conclude that we tend to see a stability in the evolution of SQ models.

Another interesting perspective is to analyze the influence of characteristics from all previous models into the ISO 25010 model. Based on characteristic naming, we put in correspondence the factors of the models. More precisely: considering a given SQ model with characteristics and their subcharacteristics, we associate characteristics of a second model to the characteristics of the first model. We do this on two steps: first consider the naming of the characteristics, then a set-based similarity is computed over the set of all keywords of the characteristics (that is the set consisting of characteristic name and subcharacteristics). The higher similarity is considered in order to associate the characteristics of the two models. The evaluation presented here uses Dice similarity measure. The final score for a characteristic is the sum of all the association score. This evaluation has highlighted the following interesting remarks: the only SQ model that contains similarities for all 8 characteristics of ISO 25010 is the McCall model. Also the highest similarity score is obtained for Maintainability characteristics from this model (significantly higher than the rest). This is explained due to the fact that similarities has been found not only performing lexical comparison between characteristics, but also computing similarities between subcharacteristics of maintainability with other

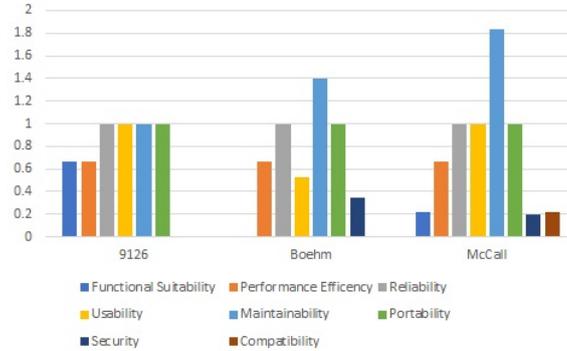


FIGURE 1. Analysis of characteristics in ISO 25010 based on similarity score

subcharacteristics of ISO 25010. The results prove the accuracy of the McCall model and also its completeness, even if it was proposed over 40 years ago and considering the dynamics of evolution of software systems.

## 6. CONCLUSIONS AND FUTURE WORK

The terminology and concepts corresponding to software quality models and standards can be characterized by a significant degree of variation. The software quality community cannot refer to a unique set of concepts and associated meanings when assessing the software quality domain or when introducing a new software quality model. As predicted by [7], since in the near future we expect a new SQ model to be proposed and standardized, an evaluation of the evolution of these models so far is an important assessment of the domain. This paper has used NLP techniques in order to determine similarity measures, which are then used to compare the evolution of consecutive SQ models and to compare the current standard, ISO 25010, with the previous ones.

The conclusions of our study can be summarized in: there are still significant modifications between models (added and removed terms), so a special care should be taken in order to avoid overloading the domain lexical set: when adding a characteristics, a survey of the previous models should be carried in order to identify if it is significantly different from the previous terms.

A second conclusion will give credits to the first SQ model introduced by McCall [12]. The similarity displayed with the current standard shows that McCall model was a very solid one, and the aspects considered in evaluating different characteristics are still valid nowadays.

As a continuation of this study, we intend to automate the process of SQ standard compliance. When a new standard will be proposed (considering that the last one, ISO 25010 was introduced in 2011, this is to be expected), software products should comply with the new standard. Instead of fulfilling all characteristics from scratch, our approach can be used to determine the set of characteristics, respectively subcharacteristics that need to be changed/added/moved or removed in order to comply the new standard.

It is also our intent to extend the SQ model evaluation to construct and ontology of SQ models, such that the evaluation can be enhanced at the graph (ontology) level using the semantic meaning of its constituents.

## REFERENCES

- [1] 9126-1, I. Software engineering – product quality. <https://www.iso.org/standard/22749.html>, 2001. Accessed: 2015.
- [2] BOEHM, B. W., BROWN, J. R., AND LIPOW, M. Quantitative evaluation of software quality. In *Proceedings of the 2nd International Conference on Software Engineering (1976)*, ICSE '76, IEEE Computer Society Press, p. 592–605.
- [3] CIANCARINI, P., NUZZOLESE, A. G., PRESUTTI, V., AND RUSSO, D. Squap-ont: an ontology of software quality relational factors from financial systems. *CoRR abs/1909.01602* (2019).
- [4] FELLBAUM, C. *WordNet – An Electronical Lexical Database*, vol. 25. 01 1998.
- [5] GOMAA, W. H., AND FAHMY, A. A. Article: A survey of text similarity approaches. *International Journal of Computer Applications* 68, 13 (2013), 13–18. Full text available.
- [6] GORDIEIEV, O., KHARCHENKO, V., FOMINYKH, N., AND SKLYAR, V. Evolution of software quality models in context of the standard iso 25010. In *Proceedings of the Ninth International Conference DepCoS-RELCOMEX, Advances in Intelligent Systems and Computing* (2014), pp. 223–232.
- [7] GORDIEIEV, O., KHARCHENKO, V., AND FUSANI, M. Software quality standards and models evolution: Greenness and reliability issues. In *Information and Communication Technologies in Education, Research, and Industrial Applications* (2016), pp. 38–55.
- [8] GRADY, R. B. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., USA, 1992.
- [9] ISO/IEC 25010:2011. Systems and software engineering. <http://www.iso.org>, 2011. Accessed: 2015.
- [10] KARA, M., LAMOUCHE, O., AND RAMDANE-CHERIF, A. Ontology software quality model for fuzzy logic evaluation approach. *Procedia Computer Science* 83 (2016), 637 – 641. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops.
- [11] KAYED, A., HIRZALLA, N., SAMHAN, A. A., AND ALFAYOUMI, M. Towards an ontology for software product quality attributes. In *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services* (Washington, DC, USA, 2009), ICIW '09, IEEE Computer Society, pp. 200–204.
- [12] MCCALL, J., RICHARDS, P., AND WALTERS, G. Factors in software quality. *Nat Tech.Information Service* 1 (1977), 0–0.

- [13] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013* (2013), Y. Bengio and Y. LeCun, Eds.
- [14] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. 2013, pp. 3111–3119.
- [15] MOTOGNA, S., LUPSA, D., AND CIUCIU, I. A NLP approach to software quality models evaluation. In *On the Move to Meaningful Internet Systems: OTM 2018 Workshops - Confederated International Workshops: EI2N, FBM, ICSP, and Meta4eS 2018, Valletta, Malta, October 22-26, 2018, Revised Selected Papers* (2018), C. Debruyne, H. Panetto, W. Guédria, P. Bollen, I. Ciuciu, and R. Meersman, Eds., vol. 11231 of *Lecture Notes in Computer Science*, Springer, pp. 207–217.
- [16] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *EMNLP* (2014), vol. 14, pp. 1532–1543.

BABEȘ-BOLYAI UNIVERSITY, DEPARTMENT OF COMPUTER SCIENCE, 1 M. KOGĂLNICEANU STREET, 400084 CLUJ-NAPOCA, ROMANIA

*Email address:* [dana@cs.ubbcluj.ro](mailto:dana@cs.ubbcluj.ro), [motogna@cs.ubbcluj.ro](mailto:motogna@cs.ubbcluj.ro), [oana@cs.ubbcluj.ro](mailto:oana@cs.ubbcluj.ro)