User-adaptable Natural Language Generation for Regression Testing within the Finance Domain

Daniel Braun[a], Anupama Sajwan and Florian Matthes
Technical University of Munich, Department of Informatics, Munich, Germany

Keywords: Natural Language Generation, Regression Testing, Finance.

Abstract: Reporting duties and regression testing within the financial industry produce huge amounts of data which has to be sighted and analyzed by experts. This time-consuming and expensive process does not fit to modern, agile software developing practices with fast update cycles. In this paper, we present a user-adaptable natural language generation system that supports financial experts from the insurance industry in analysing the results from regression tests for Solvency II risk calculations and evaluate it with a group of experts.

1 INTRODUCTION

Companies within the finance industry, like banks and insurance companies, have to fulfil many regulatory requirements. Some of the most prominent directives within the European Union (EU) include "Basel III" for banks and "Solvency II" for insurance companies. In order to fulfil the requirements introduced by these legislations, companies have to continuously report risk relevant corporate results and investments to their respective regulatory authority. These reports determine how much money companies have to put aside as a security.

From a company's perspective, it is desirable to keep this amount as low as possible, because they only can create profit from money which they can actively invest. Therefore, big insurance companies use tailored internal risk models instead of the standard risk model provided by Solvency II. The software which runs these internal risk models has to be updated regularly, in order to meet the regulatory requirements and the company's interests. Before a new version of such a software is put into production use, regression testing is used to ensure proper behaviour. A single run of such regression tests produces thousand of numbers which have to be compared to previous results and interpret by financial experts, which then have to report back to developers. This is a cost and time-intensive process which also hinders companies to deploy updates more often.

In this paper, we present a natural language generation (NLG) system which creates textual reports for the results from regression tests for Solvency II risk capital calculations. By identifying and highlighting salient patterns within the results, we want to support the work of financial experts and speed up the process. Moreover, the system is built in a way which aims to empower expert users, which are non-programmers, to adapt the system regarding the analysis which is conducted but also regarding the textual representation of the results of the analysis. The system was designed and evaluated with financial experts from a major international insurance company.

2 RELATED WORK

Most transactions on the international financial markets are nowadays not only executed but also triggered by machines. (Banulescu and Colletaz, 2013) Therefore, the relevant data is available in a machine-readable format and, due to the nature of the domain, mostly numerical. Given these circumstances, it is no surprise that the finance domain is of great interest to the NLG community, from a scientific and a commercial perspective.

One of the most prominent applications of NLG within the finance domain today is robot journalism. Together with weather, traffic, and sports, finance is one of the most popular domains for robot journalism. (Dörr, 2016) Examples for such systems were build by Kukich (1983), and more recently Liu et al. (2004), Haarmann and Sikorski (2015), Murakami

[a] https://orcid.org/0000-0001-8120-3368

Braun, D., Sajwan, A. and Matthes, F.
User-adaptable Natural Language Generation for Regression Testing within the Finance Domain.
DOI: 10.5220/0009563306130618
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 613-618
ISBN: 978-989-758-423-7

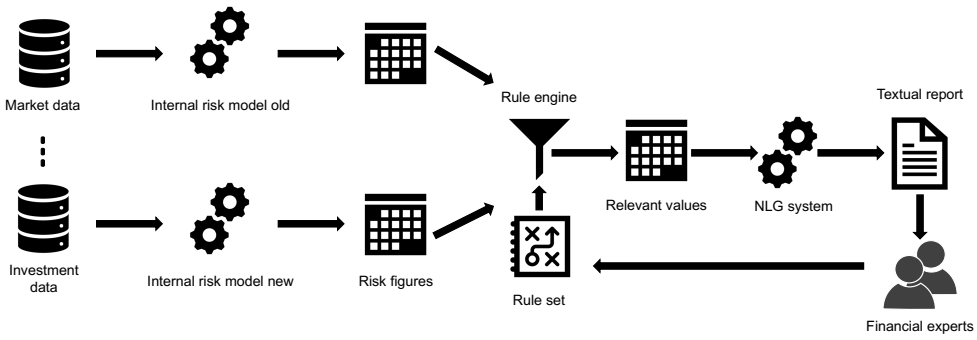

Figure 1: Worfklow.

et al. (2017), and many others. Another popular use case for NLG within the finance domain is question answering. Plachouras et al. (2016) e.g. presented a generic approach for English and Altinok (2018) focused on questions regarding financial products and services in German language. A similar system in Japanese was presented by Okuda and Shoda (2018).

3 WORKFLOW

As mentioned before, the goal of the system is to support financial experts in their daily work. Hence, we did not only try to build an isolated NLG system but to integrate it into the existing workflow at our partner company. In order to achieve this, we first interviewed financial experts from the relevant division, in order to find out how they work. The current workflow is a mostly manual process: once a new version of the software for risk capital calculations is ready for testing, both versions, the current one and the new one, are run on the same data. Each of the versions produces an Excel file with risk figures as output. Financial experts will go manually through these sheets afterwards and report relevant deviations for calculation instruments to the developers. If and when a deviation is classified as "relevant" is a non-codified process which is based on the knowledge and experience of the experts.

Based on the current worlflow, we deloped a goal workflow, which we wanted to implement (cf. Figure 1). Instead of manually checking the output of the regression testing runs, relevant patterns should be detected automatically. In order to make this possible, the knowledge about how to identify these patterns, which was previously non-codified, has to be made explicit. In the interviews with the experts, we found out that these rules are not stable, but develop over time and also might need to be adapted depending on the current run.

While the experts are familiar with mathematical formalisms and abstract thinking, they are usually not "programmers", i.e. they are usually maybe familiar with e.g. Excel formulas, but not Java or other complex programming languages. In order to create a workflow which empowers these experts to conduct the analysis on their own, we want to make use of a rule engine which enables them to modify the rules for the analysis without the help of developers. Moreover, this makes the rules more transparent and comprehensible to the experts, which will hopefully increase their trust in the system. Once the relevant patterns have been identified by the rule engine, the NLG system should generate a comprehensive report of the identified patterns. It is important to point out that the goal of the system is not a complete automation. I.e. the target audience of the generated reports are not developers but the financial experts. In this way, instead of having to browse through the whole table, financial experts can focus their valuable time on analysing and assessing identified patterns. The goal of the company is not only to save time and hence money, they also want to change their development to a more "agile" practice, which will lead to more frequent updates. With the old workflow, it would simply not be possible for the experts to keep up with the new development practice. With the proposed new workflow and the NLG system as support, we want to enable the financial experts to do so.

4 ARCHITECTURE

In order to make the system flexible and its parts reusable, we adopted a service-oriented architecture. The rule engine and the NLG system operate independently from each other. Moreover, we implemented an independent converter which pre-processes the input for the rule engine, e.g. from the Excel format.

One of the decisions we had to make was which

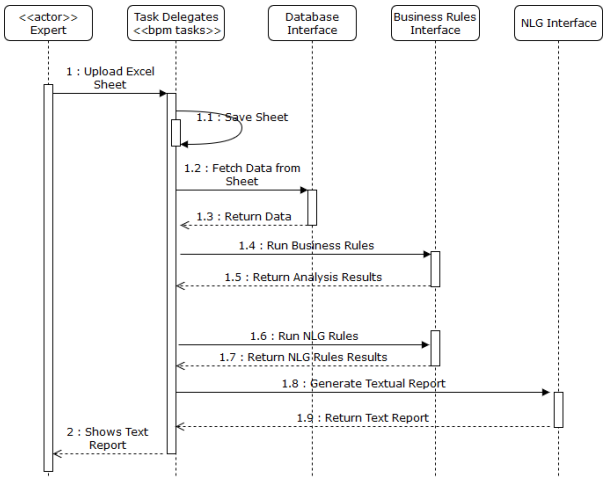

Figure 2: Sequence Diagram for Camunda.

rule engine to use. One of the most popular available rule engines is the open source software Drools (Proctor, 2012). In interviews with different employees from the insurance company, we found out that there had been previous unsuccessful attempts to use Drools in the department. People were mostly hostile towards Drools, because of its complexity and steep learning curve. Therefore, we decided to use an alternative.

The Camunda BPMN Workflow System (cf. e.g. Lammert (2015)) is already used at the department and offers a Decision Model and Notation (DMN) engine, which is expressive enough for our application and offers a graphical user interface which can be used to create, delete or edit rules. Moreover, we can use the Camunda BPMN Workflow System to orchestrate and connect our services with a web interface. The sequence diagram in Figure 2 shows in which order the different services are called by Camunda. Once the execution of the DMN engine is finished, the result is passed on to the NLG system in JSON format. A more detailed description of the rules and their output is given in Section 5. The NLG system itself internally follows the three-stage pipeline architecture described by Reiter and Dale (1997) and uses SimpleNLG (Gatt and Reiter, 2009) as surface realiser.

While the system currently only produces English output, it could be easily adapted to other languages for which SimpleNLG is available: German (Braun et al., 2019), French (Vaudry and Lapalme, 2013), Italian (Mazzei et al., 2016), Spanish (Ramos-Soto et al., 2017), Dutch (de Jong and Theune, 2018), Mandarin (Chen et al., 2018), and Galician (Cascallar-Fuentes et al., 2018).

5 USER-ADAPTABLE RULES

```
1 {
2   "category": "interest-related",
3   "number": 300,
4   "affected": 100,
5   "quantifier": "some",
6   "state": "affected",
7   "max_deviation": 120
8 }
```

Listing 1: Example result from rule execution.

The first and most important goal of the user-generated rules is the identification of relevant patterns of deviation within the results of the regression testing runs. Each calculation which is performed by the risk calculation algorithm has different categorisations, like “interest-related” or “Monte Carlo scenario”. Patterns of interest are in general multiple deviations within the same category. With the Camunda DMN engine, the experts are e.g. able to set tolerance thresholds for deviations in different categories or define the minimum number of affected values which constitute a pattern. Moreover, the experts are also able to influence the language generation on a limited level. They can e.g. use the DMN annotation to define quantifiers, like the phrase “a few” will be used if up to 15% of the values in one category are affected, “all” if 100% are affected and so on. They could also use it to express the state of affection, e.g. if the deviation is more than 5% it could be defined as “affected”. Listing 1 shows an example of how the output of a rule execution could look like. For each category which is defined as part of a rule, such a JSON object will be created.

Asset type: Synthetic Instrument
No pattern found.

Asset type: Equity Forward
No pattern found.

Asset type: Market Index
There are 10 market indexes in the database. 10 market indexes are affected. This amounts to 100.00% of all market indexes. Maximum deviation observed is 101.41%.

Figure 3: Example report from system version 1.

6 REPORT GENERATION

These JSON objects are the input for the NLG component. A first version of the system just naively created a document with a separate section for each of the JSON objects. An example output from this first version can be seen in Figure 3. This first, rather clunky version was mainly used to evaluate the contained information with financial experts (cf. Section 7). After their feedback was taken into account, a second version was developed which focused on making the text more "natural" and readable. This second version was evaluated with non-experts. Figure 4 show the output of the second version for the same input data. The new version mainly reduces redundancies by merging sentences and different categories and removing repetition of the same information.

7 EVALUATION

We ran two separate evaluations with the two different versions of the system. They only differ in the text realisation as shown in Figure 3 and 4. The first version was evaluated with a paper-based questionnaire and professionals and students from the partner company ("expert evaluation"). The second evaluation was an open online evaluation which was advertised through social media accounts ("online evaluation"). While the first evaluation focused on the tool and its utility, the second evaluation focused on the quality of the produced texts.

To the experts, the tool was presented in individual sessions, the functionality was explained and they had a chance to try the tool. Afterwards, they were shown example reports generated by the system. They were 18 participants in this evaluation which were asked to rate three statements about the shown texts on a 7 point Likert scale from "completely disagree" to "completely agree":

Asset types: Synthetic Instrument and Equity Forward
No patterns were found.

Asset type: Market Index
There are 10 market indexes in the database, all of them are affected. The results deviate up to 101.41%.

Figure 4: Example report from system version 2.

- *"The text is easy to read and understandable."* (Q1)
- *"The text is helpful for the data analysis."* (Q2)
- *"I would like to see more variation in the text."* (Q3)

Subsequently, participants were asked to rate three statements about the system which they just saw:

- *"The tool is helpful"* (Q4)
- *"The tool will help me to decide which data should be analysed deeper"* (Q5)
- *"I would like to use the tool for future analyses."* (Q6)

The results for all statements are shown in Figure 5.

In the online evaluation, participants were shown example texts from the second version of the system and three statements, which again could be rated on a seven-point Likert scale. Moreover, there was also a free text field for comments. The three statements were:

- *"The text is easy to read and understandable."* (S1)
- *"I would like to see more variation in the text."* (S2)
- *"The text is grammatically correct."* (S3)

In this evaluation, 21 people participated, the results are also shown in Figure 5.

Overall, both evaluations have been very positive, especially the expert evaluation. The only clear negative reaction was from the online evaluation, where participants did not find the texts easy to read and understandable. Five participants also mentioned in the free text comment that they struggled to understand the texts. Two participants mentioned that they did not even understand the words which were used. Looking at the example in Figure 4, it is not surprising that non-experts struggle to understand the texts and it is not necessarily a shortcoming of the system. While in general positive, the meaningfulness of the

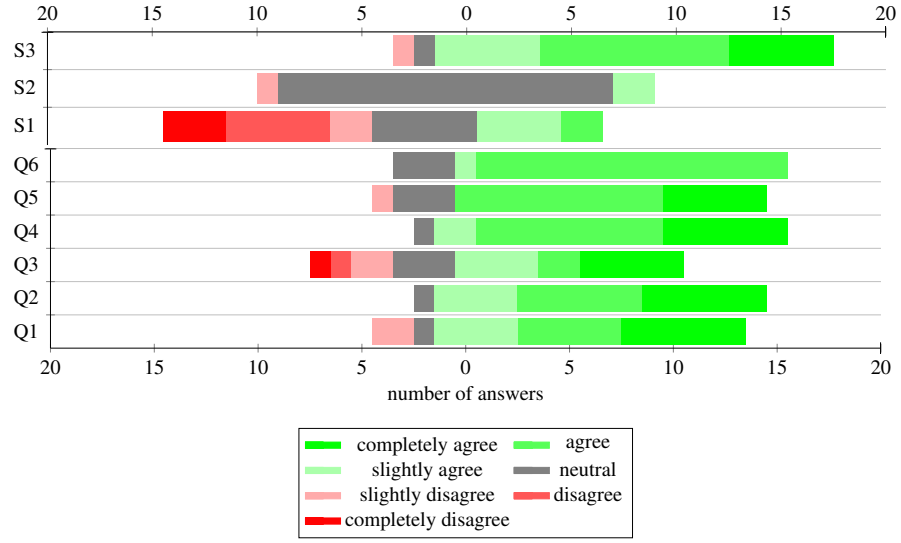

Figure 5: Likert scale results from the evaluation.

results for Q3 and S2 is rather limited, since participants only saw a few texts in a very short time and it might still be that they get "tired" of repetitions when using the system for a longer time.

8 CONCLUSION

In this paper, we presented a system that supports financial experts from the insurance industry to evaluate the results of regression testing runs for Solvency II risk calculations. We integrated the system in the existing workflows and empowered the users to adapt the rules by which the system operates. A first evaluation with expert users showed promising results. In the future, we would like to conduct a task-based evaluation to check, whether the positive perception of the system is also matched by a measurable improvement in task solving.

REFERENCES

Altinok, D. (2018). An ontology-based dialogue management system for banking and finance dialogue systems. *arXiv preprint arXiv:1804.04838*.

Banulescu, D. and Colletaz, G. (2013). High&frequency risk measures.

Braun, D., Klimt, K., Schneider, D., and Matthes, F. (2019). Simplenlg-de: Adapting simplenlg 4 to german. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 415–420, Tokyo, Japan.

Cascallar-Fuentes, A., Ramos-Soto, A., and Bugarín Diz, A. (2018). Adapting SimpleNLG to Galician language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 67–72, Tilburg University, The Netherlands. Association for Computational Linguistics.

Chen, G., van Deemter, K., and Lin, C. (2018). SimpleNLG-ZH: a linguistic realisation engine for Mandarin. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 57–66, Tilburg University, The Netherlands. Association for Computational Linguistics.

de Jong, R. and Theune, M. (2018). Going Dutch: Creating SimpleNLG-NL. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 73–78, Tilburg University, The Netherlands. Association for Computational Linguistics.

Dörr, K. N. (2016). Mapping the field of algorithmic journalism. *Digital Journalism*, 4(6):700–722.

Gatt, A. and Reiter, E. (2009). Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93. Association for Computational Linguistics.

Haarmann, B. and Sikorski, L. (2015). Natural language news generation from big data. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(6):1454–1460.

Kukich, K. (1983). Design of a knowledge-based report generator. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, pages 145–150. Association for Computational Linguistics.

Lammert, J. (2015). Evaluation des camunda bpmn workflow systems. Master's thesis, Universität Magdeburg.

Liu, Q., Lu, X., Ren, F., and Kuroiwa, S. (2004). Automatic estimation of stock market forecasting and generating the corresponding natural language expression. In *Information Technology: Coding and Com-*

puting, 2004. Proceedings. ITCC 2004. International Conference on, volume 1, pages 241–245. IEEE.

Mazzei, A., Battaglino, C., and Bosco, C. (2016). SimpleNLG-IT: adapting SimpleNLG to Italian. In *Proceedings of the 9th International Natural Language Generation conference*, pages 184–192, Edinburgh, UK. Association for Computational Linguistics.

Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H., and Miyao, Y. (2017). Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1374–1384.

Okuda, T. and Shoda, S. (2018). Ai-based chatbot service for financial industry. *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL*, 54(2):4–8.

Plachouras, V., Smiley, C., Bretz, H., Taylor, O., Leidner, J. L., Song, D., and Schilder, F. (2016). Interacting with financial data using natural language. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1121–1124, New York, NY, USA. ACM.

Proctor, M. (2012). Drools: A rule engine for complex event processing. In *Proceedings of the 4th International Conference on Applications of Graph Transformations with Industrial Relevance*, AGTIVE'11, pages 2–2, Berlin, Heidelberg. Springer-Verlag.

Ramos-Soto, A., Janeiro-Gallardo, J., and Bugarín Diz, A. (2017). Adapting SimpleNLG to Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 144–148, Santiago de Compostela, Spain. Association for Computational Linguistics.

Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Vaudry, P.-L. and Lapalme, G. (2013). Adapting SimpleNLG for bilingual English-French realisation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 183–187, Sofia, Bulgaria. Association for Computational Linguistics.