SUPPORTING KNOWLEDGE RE-USE WITH EFFECTIVE SEARCHES OF RELATED ENGINEERING DOCUMENTS - A COMPARISON OF SEARCH ENGINE AND NATURAL LANGUAGE PROCESSING-BASED ALGORITHMS

Arnarsson, Ivar Örn (1); Frost, Otto (2); Gustavsson, Emil (2); Stenholm, Daniel (1); Jirstrand, Mats (2); Malmqvist, Johan (1)

1: Chalmers University of Technology; 2: Fraunhofer-Chalmers Centre

ABSTRACT

Product development companies are collecting data in form of Engineering Change Requests for logged design issues and Design Guidelines to accumulate best practices. These documents are rich in unstructured data (e.g., free text) and previous research has pointed out that product developers find current it systems lacking capabilities to accurately retrieve relevant documents with unstructured data. In this research we compare the performance of Search Engine & Natural Language Processing algorithms in order to find fast related documents from two databases with Engineering Change Request and Design Guideline documents. The aim is to turn hours of manual documents searching into seconds by utilizing such algorithms to effectively search for related engineering documents and rank them in order of significance. Domain knowledge experts evaluated the results and it shows that the models applied managed to find relevant documents with up to 90% accuracy of the cases tested. But accuracy varies based on selected algorithm and length of query.

Keywords: Natural Language Processing, Semantic data processing, Machine learning, Knowledge management

Contact:
Arnarsson, Ivar Örn
Volvo Trucks / Chalmers
Product Development
Sweden
varo@chalmers.se

Cite this article: Arnarsson, I.Ö., Frost, O., Gustavsson, E., Stenholm, D., Jirstrand, M., Malmqvist, J. (2019) 'Supporting Knowledge Re-Use with Effective Searches of Related Engineering Documents - A Comparison of Search Engine and Natural Language Processing-Based Algorithms', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.266

1 INTRODUCTION

During a product development (PD) project, product developers often need to make changes to previous designs in order to improve, enhance or adapt a product to identified opportunities. Eckert *et al.* (2000) said that when designers are confronted with a new design, their starting point is to look for old designs, often the current design version or reusing other existing designs for components. The novelty introduced by this paper is to automate that search of the designers so that they do not spend time searching for information until they find a suitable source (Eckert *et al.*, 2000). Information about product changes, including structured data (e.g., numerical data, categorical data and timestamps) mixed with unstructured data (e.g., text inserted by engineers), is stored in databases to keep track of evolving design solutions, verification and decision documents (Pikosz & Malmqvist, 1998). Large PD projects can contain tens of thousands of product change documents (Arnarsson *et al.*, 2017), also known as Engineering Change Requests (ECRs) (Jarratt *et al.*, 2011).

Further, within a project-oriented organization there is commonly a process to capture lessons learned (LL) from each and every project to accumulate best practices in order to continuously improve upcoming projects. This process of codifying best practices is commonly resulting in documents referred to as Design Guidelines (DGs).

Moreover, these two document types have different goals, ECRs focusing on achieving a rapid process from the identification of issues until resolution and DGs focusing on creating a reusable asset, decreasing the risk of repeating the same mistake again and establishing best practice guidelines. ECRs and DGs are stored in separate databases; overview of the processes within PD can be seen in Figure 1.

With the fast development of data collection and storage, companies are now faced with large volumes of complex data from multiple sources (Wu *et al.*, 2014). Arnarsson *et al.* (2017) have identified the need of product developers to perform advanced searches within and across several databases for efficient problem investigation and resolution. The problems have been that i) search functionality is limited to only considering structured data and not taking into account the rich information stored in unstructured data (e.g., text data) and ii) searching across multiple databases simultaneously has not even been a practical option. Current search results (in single databases) return up to hundreds of reports back when filters are applied, such as timeframe, vehicle types and keywords. It follows that it is time-consuming to identify the most relevant documents.

It is therefore important to be able to connect current PD data, e.g. knowledge management systems containing LL from historical projects and product design change documents. Furthermore, to identify structurally related information on such items as same part or contextually related documents, dealing with similar issues or problems based on structured and unstructured data, short list of the most relevant documents is produced.

Recent developments in data mining and Natural Language Processing (NLP) have made it possible to search through huge databases of text documents to retrieve relevant items based on variety of search queries. By utilizing NLP methods such as search engine methodology or document embedding approaches, one can find documents (in this case ECRs and DGs) that are similar to each other by looking at the context of sequential words together, instead of isolated words. Previously, simpler methods relied on the notion that related documents share the exact same words and to find similar documents, one usually searches through the document set to find the ones that match best according to word count. More elaborate methods, such as doc2vec-methodology (Le & Mikolov, 2014), can understand the context of a document and automatically detect synonyms to words utilized.

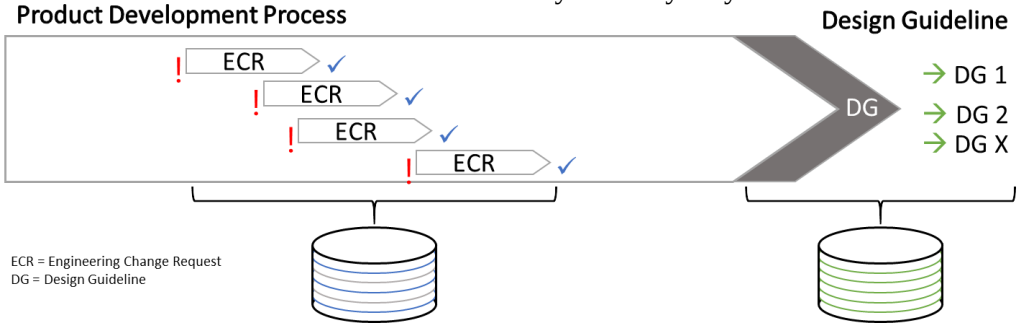

Figure 1: Product development process displaying the database capture of ECRs and DGs.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

Another benefit of these types of NLP methods is that they provide a similarity metric between documents, meaning that one can get a numeric value quantifying the degree of similarity between pairs of documents. This metric can be utilized for performing correlation analyses (Steinbach *et al.*, 2000) where one tries to group together related documents into clusters.
This paper explores the research gap of performing advanced searches based on NLP techniques both within a single database and across several engineering databases. Such models would help product developers assure better pre-studies as they could now evaluate a short list of the most relevant historical documents that might contain valuable knowledge.
The correlation analysis can be utilized for summarizing and grouping together similar issues that occurred during the PD project.
The aim of the work outlined in this paper is thus to develop and test search queries that can be further used for correlating documents in two engineering databases. Therefore, relationship between documents that will support engineers in making better decisions in future projects were identified. Specifically, we addressed two research questions:

1. *What type of search algorithm should be used for finding relevant information from both Engineering Change Requests & Design Guideline documents?*
2. *What kind of useful information can product developers obtain from the document embedding approach?*

The remainder of the paper is structured as follows: Section 2 presents earlier work, Section 3 outlines research approach, Section 4 presents results, while Section 5 summarizes a discussion and Section 6 concludes the paper and discusses ideas for future research agenda.

2 EARLIER WORK

Let us now review earlier work on the data information needs of product developers, text analyses and NLP models. Arnarsson *et al.* (2017), interviewed twenty individuals with diverse experience throughout the PD process in order to identify the information needs of product developers. They identified that there is an opportunity to perform integrated searches across multiple databases that should include structured and unstructured data; however, in the case explored this was not currently accomplished. Current IT systems have limited ability to search unstructured data but interviews unveiled that most of the knowledge documented is available in the form of unstructured data including report title, description of changes, actions taken, comments and technical solutions. Further, the documents of interest should then be correlated and clustered together in order to identify the related knowledge residing in various documents. Clarkson *et al.* (2004) looked into the prediction of changes in complex designs and stated that design changes can be connected to manage risk and evaluate redesigns.
The ECR process is part of the engineering change process, corresponding to the change trigger approval states described by Jarratt *et al.* (2011) in their generic engineering change management process. The engineering change process takes place under different PD phases and various authors have proposed generic models. Eckert *et al.* (2004) divided the sources of change throughout the design process into two categories: emergent changes and initiated changes. Jarratt *et al.* (2011) suggested a six-state engineering change process that begins with the ECR to be raised, the identification of solutions, a risk assessment of solution(s), selection and approval of a solution, implementation and, finally, a review of the change.
A DG represents knowledge or understanding resulting from either a positive or negative experience which has been captured and resulted in a proposed way of action (Weber *et al.*, 2001). The literature reports various formats and capture techniques for DGs in order to efficiently become applied in future development projects. A few common techniques include LL sessions, post-action reviews, project debriefing, post-project reviews, Postmortems and Engineering Checksheets (Schindler and Eppler 2003; Catic and Malmqvist 2013). However, the process of identifying which lessons that should be captured is often non-systematized and irregularly performed (Chirumalla 2014).
The extraction of design and manufacturing text content has previously been performed successfully using NLP and node models by e.g. Dong & Agogino (1997), Catron & Ray (1991) and Kim & Wallace (2009). Dong (2005) further explored design team communication documents using a latent semantic approach. In more general terms, the development of NLP methods for summarizing and interpreting entire documents have been increasing over the past few years. Methods for transforming

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

single words into high-dimensional representations have previously been extensively studied (see, e.g., word2vec, Mikolov *et al.* (2013)). Le & Mikolov (2014) developed this notion further by introducing a concept called doc2vec, which is word embedding methodology where one trains a model to translate entire documents into a high-dimensional numerical representation. This numerical representation of documents can be utilized for comparing different documents, for finding similar documents, and for clustering/grouping documents into different themes. Two powerful properties that can be achieved by methods like doc2vec include; a) contextual information which can in some cases be interpreted and b) synonyms to words utilized in a document which can automatically be encoded in numerical representations.

3 RESEARCH APPROACH

The study is based on two PD databases containing ECRs and DGs from real commercial vehicle PD projects within an organization. The NLP search method enables a way to correlate data within and between the two databases. We therefore focus on unstructured data as previous findings suggest that current IT systems are deficient in searching and retrieving relevant documents based on text data. Figure 2 below describes the methodology and process used in this research. The data from the two databases can flow into two streams representing the Elasticsearch (search engine) and document embedding models (DEMs) that were tested. The DG database has a more complex scheme as the format of those documents is structured more loosely and therefore need more complex data extraction, represented by the "DG Data extraction" box. The two models then have a common interface denoted "Search service".

The setup of the research project as a partnership with the case company gave access to a detailed inquiry about the topic in a real setting. This access to data was the main rationale for selecting the single-company design, i.e., an opportunity to study a situation otherwise inaccessible to researchers, which Yin (2013) refers to as a "revelatory case".

The ECR data contained 7,732 documents from recent development projects. Each ECR is made up of more than 50 variables that are used to document and follow up on the logged changes. The documents are log files of design and test-related issues that need to be resolved. They contain variables such as: title, part name, problem description, root cause, solution and test results.

The database of collected design guidelines is structured into a spreadsheet where each row contains a 'good-to-know' item further referred to as Knowledge Element (KE). Each KE contains a rationale including Declarative knowledge (Know-What to do), Procedural knowledge (Know-How to do it) and Causal knowledge (Know-Why it should be done) (Alavi and Leidner, 2001; Lundvall and Johnson, 1994). Additionally, each KE can include illustrative images and references to other sources or individuals that can support the reuse of knowledge. In total 25 DGs were analysed in this study and together they make up a total of over 800 KEs.

The search application considered in this project consists of two modules; the Elasticsearch and the DEM. These modules are tied together by a front facing search service that can pass queries to both the Elasticsearch and DEM.

For the DEM (i.e., the model that translates each document into a high-dimensional numerical representation), a pipeline consisting of three steps was used. First, the data cleaning and normalization step reduced the cardinality of the set tokens in the data set, reducing the data set complexity to improve the robustness and accuracy of the final model. This included lowercasing, removing non-letter characters and lemmatization using the WordNet lemmatizer (Bird and Loper, 2004). Second, the DEM step trained a doc2vec model (Le & Mikolov, 2014) using the gensim library (Rehurek & Sojka, 2010) on the normalized data. Third, this model was used by the search service that given a query, transformed the query into a document vector that could be matched against the embedded documents of the model.

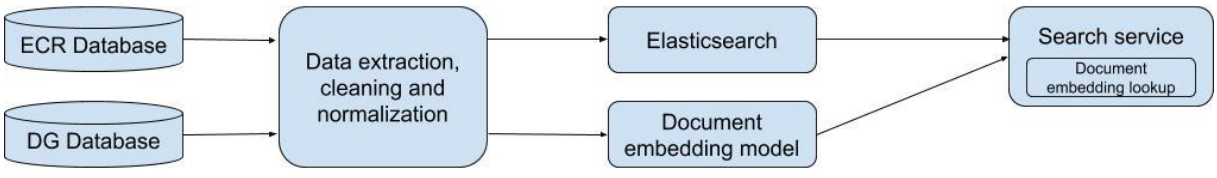

Figure 2: Process of methodology used where two databases Elasticsearch and DEM are fed into two streams and tested with different models.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

The similarity measure used for finding similar documents was computed as the cosine distance (angle between the two high-dimensional vectors) and the score was a number between -1 and 1 where 1 indicated a perfect match and -1 indicated no match at all.
For the search engine (i.e., standard search methods that rely on matching exact words), the pipeline is somewhat simpler. The search engine used was Elasticsearch (2018) that comes with all the functionality required to build a search-based application. Similar to the normalization step in the DEM, the input data to the model was lowercased, but additionally, all the stop words were removed from the text, to remove the noise otherwise caused by these common terms. Elasticsearch is part of this project to provide some benchmark results for validating the DEM.
While Elasticsearch serves as the base line for a typical information retrieval task like the one presented in this paper, the DEM represents a novel approach to information retrieval tasks of this kind. The DEM is simple to use and because it produces a numerical representation of the text, the output can be combined with any other data mining or feature extraction method that also produces numerical data. Therefore, the DEM can be utilized not only for information retrieval tasks but also for other analytical tasks involving ECRs, such as classification or clustering.
The results were then validated together with domain knowledge experts from the company, as the text in documents is a unique case. A domain knowledge expert validated the outcome of the results queried results. The validation criteria of acceptance were set to three levels; green represented a good relation to part and failure mode, yellow indicated a good relation to either part or failure mode, whereas red indicated no relation to part or failure mode. This research will therefore evaluate the possibility of finding and analysing correlations between Engineering Change Request and Design Guideline databases. Our focus was on evaluating whether the results were relevant without comparing score values.

4 RESULTS

4.1 Example of query & result

In the study there were 7,732 documents available to query on and score against.
Below is a query example (Figure 3) where an entire document is queried to the DEM and the top three most similar documents are listed below. The document-embedded model has a score from -1 to 1, where 1 means the two documents match perfectly. The score is the cosine distance between document vectors and should not be compared internally between similar documents from the results but rather for evaluating the ranking order of similarity to the queried document. The grey highlighted row is used as input to the specific query. This was the format of the result on which domain knowledge experts performed their evaluations.
The Elasticsearch minimum score is 0 but it is hard to give a maximum value as the scores for the results from Elasticsearch have a complex data dependency. The scores are used for sorting the results since it is hard to compare the scores between different searches. However the maximum score retrieved for Elasticsearch generated was 72.24.

Case	Title	Description	Score
90304	Difficult to push the amplifier branch through main hole	Fault/problem/cause/background During harness routing, we have to push amplifier branch through a small hole. This operation is damaging to wires and time consuming, the hole is too small. Action: Enlarge hole on amplifier side. Secure the harness by changing the clipping point.	1
92388	Not enough room to push sleeper harness in sleeper main hole	Fault/problem/cause/background When routing the sleeper harness in living area, we have to push branches though small holes in the bunk frame. Action: We need bigger holes to not damage harnesses and win assembly time. Comments: Prime issue #258. Test: Log files attached (if SW problem):	0.531
123457	Adjustment of hole in panel needed for panel fastener	Fault/problem/cause/background Adjustment of hole needed for kick panel plastic fastener Action: Add a ~5mm to diameter of hole for the plastic fastener for part 800001 to allow the plastic pin to come through.	0.454
51345	Difficult to fit harness through pillar	Fault/problem/cause/background During harness routing, we have to "Fish" a branch through a small hole on pillar. This operation would need a special tool due to bad ergonomic position and take too much time. Action: Change harness rout to pillar.	0.454

Figure 3: Example of the format of results when the DEM is being queried by an entire document.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

4.2 Query search service

We performed an evaluation and tested a search application keeping in mind that evaluation criteria's depend on the use case and needs of every individual user. To test the results from the queries, 40 documents were queried with the DEM and similarly for the Elasticsearch; 20 with short queries of a few words and 20 with entire documents queried. We limited ourselves to querying 40 documents due to the fact that the top three results for each query had to be manually evaluated for each model, making it 240 results in total for domain knowledge experts to evaluate. The two sets of 20 documents were queried with both models for comparison. The top three scores from each query were logged into a matrix (Tables 1 & 2) together with the number of words queried. Experts with domain knowledge validated the resulting matrices below by giving each score a green, yellow or red colour. For the search application presented in this paper, we have also performed some exploratory testing where experts simultaneously tested and evaluated the resulting scores. Therefore, we have been able to draw the following conclusions.

Elasticsearch gives the user very powerful means for finding documents with high precision (i.e., documents that are very relevant according to a domain expert); see Tables 1 & 2. The downside, however, is that the user must have a relatively good knowledge of the documents he/she is searching for. Additionally, the queries required to produce high quality results might be far from trivial, requiring the user to possess the knowledge to produce such queries. We believe that the Elasticsearch method is appropriate when the user knows, in detail, what documents he/she is looking for (e.g., when the user knows that a specific word is very important and that other words should be discarded). An example is when one searches for information regarding "engine stop" as a failure mode. Then many results regarding engine stop as an initial test mode will also appear with the consequences that a well-educated user would specify the search query as "engine stop NOT initial test mode". However, for more complex situations, this is not as straightforward. The DEM provides an easy-to-use free text query format that produces relevant results even for short queries (i.e., queries of only few words). The downside of this method is mainly that the model is opaque in the sense that it is both random (the training of the model depends on random shuffling of the data) and hard to explain. For example, if the user inputs a query and finds that a subset of the results is not relevant it is not obvious how to extend the query in such a way that the unwanted subset is excluded. Whereas a possibility of providing negative weights as inputs to the DEM exists, our initial findings indicate that doing so has a negative impact on results.

Regarding execution time of the two search methods, the time it takes to perform a search is sub-second for both when considering a database containing approximately 10,000 documents. The DEM will not scale as well as the Elasticsearch when the size of the databases increases. However, for ECR database sizes, the execution time will never become an issue.

We defined a successful relation for a case when there was at least one green score together with one yellow out of three scores. There is no guarantee that there would be a match to each case, which could explain some of the no relation (red) marks.

4.2.1 Search based on short queries

For the short (less than six words) free text queries, we evaluated both the DEM and Elasticsearch. We found that in both cases we were able to find green or yellow scored documents, the DEM retrieving matches for around 50% of queries, whereas Elasticsearch scored 90%. DEM results showed more red cells (no relation) than Elasticsearch but there were also cases where we managed to retrieve three scores with good relation to the queried document. DEM and Elasticsearch results for short queries can be seen in Table 1.

4.2.2 Search based on long queries

For search a query that includes the entire content of a document, domain knowledge experts evaluated the DEM to be able to find similar ECRs given ECR or DG in about 35% of test cases and 20% for Elasticsearch. It is harder to use a simple search engine (Elasticsearch) for long queries since the user would need to translate a document into a relevant search query. Document embedding model and Elasticsearch results for documents that were fully queried can be seen in Table 2.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

Table 1: Matrix showing the results of 20 random queries with a few words using the DEM and Elasticsearch, respectively. Displaying case number, number of words queried and top three scores for each query

		Document Embedding Model			Elasticsearch		
Case	**Number of Words**	**Score 1**	**Score 2**	**Score 3**	**Score 1**	**Score 2**	**Score 3**
1126	4	0.47	0.45	0.34	24.30	17.05	15.46
229	3	0.48	0.38	0.38	11.02	8.69	8.51
1116	6	0.51	0.51	0.48	30.12	25.16	15.49
1727	3	0.58	0.57	0.57	18.83	14.09	14.04
1115	3	0.65	0.63	0.59	15.91	15.46	14.97
60	4	0.55	0.53	0.52	20.27	19.70	17.96
1822	2	0.62	0.61	0.57	10.07	9.76	9.11
1117	5	0.55	0.54	0.53	11.73	10.37	10.37
301	2	0.61	0.60	0.54	15.38	12.12	10.90
1905	3	0.61	0.59	0.56	17.16	14.41	14.36
1293	2	0.66	0.63	0.63	14.97	14.97	13.17
334	3	0.44	0.42	0.41	11.39	9.99	9.82
1864	5	0.59	0.57	0.57	17.13	16.38	15.53
1853	4	0.63	0.62	0.60	16.84	13.09	12.75
673	3	0.60	0.59	0.58	15.88	15.51	15.51
1397	3	0.68	0.54	0.52	15.64	11.87	11.64
1120	4	0.61	0.58	0.57	25.28	21.14	11.38
3481	4	0.52	0.51	0.50	24.30	20.28	18.77
2119	3	0.64	0.61	0.61	16.59	14.88	14.62
3440	3	0.51	0.50	0.49	13.59	10.40	8.86

Good relation
Some relation
No relation

Table 2: Matrix showing the results of 20 random queries using entire text of documents for the DEM and Elasticsearch, respectively. Displaying case number, number of words queried and the top three scores for each query.

		Document Embedding Model			Elasticsearch		
Case	**Number of Words**	**Score 1**	**Score 2**	**Score 3**	**Score 1**	**Score 2**	**Score 3**
353	92	0.47	0.46	0.46	39.51	39.51	38.09
601	75	0.59	0.52	0.51	43.25	42.59	41.34
748	44	0.52	0.51	0.51	50.03	39.51	38.34
1302	80	0.51	0.42	0.40	46.91	38.38	32.76
1720	95	0.40	0.38	0.37	42.04	39.51	39.50
2275	9	0.65	0.59	0.58	72.24	40.46	39.51
3260	101	0.46	0.46	0.46	39.51	37.43	36.60
3656	71	0.50	0.44	0.43	52.24	39.51	39.51
3720	73	0.51	0.44	0.42	61.03	55.16	40.30
3960	91	0.43	0.43	0.43	50.21	42.19	39.51
4279	63	0.42	0.42	0.40	40.52	39.51	37.43
4526	143	0.42	0.42	0.39	40.34	39.79	39.51
5925	46	0.62	0.48	0.42	48.65	47.47	43.30
5939	118	0.40	0.40	0.39	48.28	43.30	39.51
6317	53	0.47	0.47	0.46	42.57	40.95	39.99
6478	61	0.45	0.41	0.41	62.06	39.51	39.51
6869	57	0.65	0.48	0.47	66.79	66.79	41.38
7074	73	0.41	0.41	0.40	48.77	39.51	39.51
7083	165	0.40	0.38	0.38	39.51	39.51	37.43
7266	114	0.41	0.40	0.40	39.51	39.51	37.44

Good relation
Some relation
No relation

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

5 DISCUSSION

Let us now consider the stated research questions in relation to the findings and transferability of results.

5.1 RQ1: *"What type of search algorithm should be used for finding relevant information from both Engineering Change Requests & Design Guideline documents?*

The method that should be used for finding relevant documents in a database depends greatly on the type of search queries selected and on the users that will utilize the search application. If the search queries are short and concise, when one basically wants to find documents associated with very specific words or sentences, search engines such as Elasticsearch are preferred. Elasticsearch retrieved matching documents in 90% of the test cases compared to 50% for the DEM. When, however, more complex search queries are to be performed, e.g., entire documents are to be queried in order to find similar documents, a document embedding approach is slightly more suitable (35% match) than Elasticsearch (25% match). An overview of scores can be seen in Table 3 below. If a user is well educated in performing search queries, he/she should excel in formulating complex structured queries to a search engine. However, if the user is not well-educated, he/she would benefit from the document embedding approach, which is less sensitive to how the actual query is performed. The drawbacks to the DEMs are that they are basically black box models that provide a result to your search query without explicitly stating why this was the best match. For the case considered in this project where an entire document is used as a search query in order to find a group of similar documents, the DEM has shown the potential of being the appropriate choice. The best solution, however, is probably a hybrid between the two so that depending on the search query, some governing method decides which search method to employ for the specific query.

5.2 RQ2: *"What kind of useful information can product developers obtain from the document embedding approach?"*

Based on the search algorithms tested, we can identify a list of documents with ranked scores providing a measure of distance to the queried document. Similar documents were identified by researchers and domain knowledge experts through a manual inspection and evaluation of 40 documents along with the outcome of their resulting queries. It is hard to pinpoint a threshold in the score for a relation between documents since the scores for the DEM are cosine values representing a distance, whereas Elasticsearch scores differ greatly between short and long queries.

Product developers can take advantage of the rank document list to rapidly identify documents of value to the specific situation. From an ECR perspective, this helps to identify historical issues related to a current issue and ensuring better pre-studies on historical documents before making a new product or the re-design of a product. It can take product developers a couple of hours to make a manual list of related documents but using a model like this reduces that time to minutes. Given that we managed to find relevant with different precisions dependent on model type and query length, it is safe to say that we can work further with the results to verify whether the models did not retrieve more relevant documents because they might not exist. The reliability of the models can be further improved by performing feedback loops back to the model with the cases that were considered to be unrelated by experts. Then the model would learn that the connections it made previously were irrelevant.

From the DG perspective, there are multiple promising views to support increased reuse of best practices within an organization. By being able to identify relevant documents within an organization related to a specific DG, possibilities to more easily identify related knowledge are opened up in other parts of the organization. Finding duplicate of knowledge that can be merged to clean up existing codified assets and finding other relevant parties in the organization that might be interested in the DG selected.

Table 3: Matrix summarizing the successful retrieval rate for the DEM and Elasticsearch for short and long queries.

	Document Embedding Model	Elasticsearch
Short query	50%	90%
Long query	35%	25%

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

5.3 Limitations and transferability of results

The studied topic of correlation between data points within datasets of either ECRs or DGs in an engineering context called for a deeper understanding and a possibility of analysing real-life data. As mentioned in the research approach, the setup of the research project as a partnership with the case company gave access to a detailed inquiry about the topic in a realistic setting. However, the single case study research design, including the analysis of two databases, ECRs and DGs, results in limitations when it comes to transferability of results. Also, it is difficult to rigorously evaluate any search application since the evaluation criteria depend on the use case. In this study, a manual evaluation was performed on each query output resulting from the search application. Nevertheless, throughout the study there has been no evidence that the method would not be applicable to similar studies where data analysis is to be performed where the goal is to find similarity between text-rich documents. Therefore, we believe that the study can be transferred to most cases where unstructured data is accessible. The field of study should not matter since the algorithms query document text. Regarding the scalability, the DEM scales linearly with the number of documents. I.e., if we double the number of documents in the index, the time to respond to a query would double as well. Elasticsearch is built with a focus on scalability and has a distributed index which promises horizontal scalability for data up to the petabyte scale. In essence, the response time for a query is constant relative to the number of documents indexed.

6 CONCLUSIONS AND FUTURE WORK

This paper concludes that there are potential benefits to the approach of querying engineering documents by their similarities to the use of NLP models. The approach was performed with two databases. However, so far there seems to be no limitations to include additional databases into the pipeline of the search service. It can be seen that the Elasticsearch and the DEM work in some cases and not in others. By looking at Tables 1 & 2, the results show that the NLP methods can find and rank documents (two out of three documents with at least one green and one yellow mark) in a minimum of 25% of the cases and a maximum of 90% of the cases. It remains to be figured out why some cases do not work as well as others. By manually evaluating relevant documents and then querying them to see if algorithms can detect them. Another option to consider is that there are no relevant documents in the databases for a specific query and therefore, a red mark might be the reason.
Conclusions regarding the two NLP approaches are that the preferred method for finding similar documents in a database depends on the type of search queries that will dominate the application. If the search queries are short and concise and it is clear what the user would like to obtain, search engines such as Elasticsearch are much more powerful (see Table 1). If, however, the user wants to find similar documents, the document embedding procedure is more appropriate (see Table 2).
There are several promising continuations to this work. First, feed the domain knowledge expert evaluation matrix back to the models to improve the precision of the results. Second, create a demonstrator within industry to further evaluate the value of such a model to a product developer. Finally, regarding the DEM, we believe that it can be extended further to fit the specific use case intended. Connected to each ECR and DG, there is a wealth of metadata consisting of both numerical and categorical data. This project has focused on only using the free text data included in the documents, which means that future work includes analysing how it is possible to incorporate also the numerical and categorical data into the documents. By using this information when either training the DEMs (i.e., providing context to the documents) or by using the numerical/categorical data when measuring the similarity between documents (e.g., providing that documents concerning the same specific vehicle part should be estimated to each other even if their free text data is dissimilar according to the model) could improve the model. Another idea is to try to extend the recent word representation models BERT (Devlin *et al.*, 2018) and ELMo (Peters *et al.*, 2018) to also be applicable to entire documents. Both these models have been shown to outperform the word2vec models on benchmark datasets and it might be possible to extend them to entire documents. The extended versions might also perform better on a document level compared to the doc2vec methodology utilized in this project.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press

ACKNOWLEDGEMENTS

The financial support by AB Volvo and by the Swedish Governmental Agency for Innovation (Vinnova) is greatly acknowledged. This study was conducted within the Vinnova Vinnex Excellence Centre for Product Realization and the Production Area of Advance at the Chalmers University of Technology.

REFERENCES

Arnarsson, Í.Ö., Gustavsson, E., Malmqvist, J. and Jirstrand, M. (2017), "Design analytics is the answer, but what questions would product developers like to have answered?", *Proceedings of the 21st International Conference on Engineering Design (ICED 17)*, Vancouver, Canada, August 21-25, 2017

Bird, S. and Loper, E. (2004), "NLTK: the natural language toolkit. "*Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics.

Catic, A. and Malmqvist, J. (2013), "Effective method for creating engineering checklists", *Journal of Engineering Design*, Vol. 24 No. 6, pp. 453–475, available: http://dx.doi.org/10.1080/09544828.2013.766824.

Catron, B.A. and Ray, S.R. (1991), "ALPS: A language for process specification", *International Journal of Computer Integrated Manufacturing*, Vol. 4 No. 2, pp. 105–113.

Chirumalla, K. (2014), "Analyzing lessons learned practice in complex product development: Identification of barriers and recommendations", *International Conference on Intellectual Capital and Knowledge Management and Organisational Learning*, Academic Conferences International Limited, p. 108.

Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting change propagation in complex design". *Journal of Mechanical Design*, Vol. 126 No. 5, pp. 788–797.

Devlin, J., et al. (2018), "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv,1810.04805*.

Dong, A. (2005), "The latent semantic approach to studying design team communication", *Design Studies*, Vol. 26 No. 5, pp. 445–461.

Dong, A. and Agogino, A.M. (1997), "Text analysis for constructing design representations", *Artificial Intelligence in Engineering*, Vol. 11 No. 2, pp. 65–76.

Eckert, C., Clarkson, P.J. and Zanker, W. (2004), "Change and customisation in complex engineering domains", *Research in engineering design*, Vol. 15 No. 1, pp. 1–21.

Eckert, C.M., Stacey, M.K. and Clarkson, P.J. (2000, January). "Algorithms and inspirations: creative reuse of design experience". In *Greenwich 2000 International Symposium: Digital Creativity*, University of Greenwich, London, pp. 1–10.

Elasticsearch B.V., accessed 01.11.2018, https://www.elastic.co/products/elasticsearch.

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H. and Clarkson, P.J. (2011), "Engineering change: an overview and perspective on the literature", *Research in engineering design*, Vol. 22 No. 2, pp. 103–124. https://dx.doi/org/10.1007/s00163-010-0097-y

Kim, S. and K. Wallace (2009), "An Automatic Identification of Negation in Design Documents", *ICORD 09: Proceedings of the 2nd International Conference on Research into Design*, Bangalore, India 07.-09.01. 2009.

Le, Q. and Mikolov, T. (2014), "Distributed representations of sentences and documents", *International Conference on Machine Learning*.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), "Efficient estimation of word representations in vector space", *arXiv preprint arXiv: 1301.3781*.

Peters, M. E., et al. (2018), "Deep contextualized word representations", *arXiv preprint arXiv:1802.05365*.

Pikosz, P. and Malmqvist, J. (1998), "A comparative study of engineering change management in three Swedish engineering companies", *Proceedings of the DETC98 ASME design engineering technical conferences*, pp. 78–85.

Rehurek, R. and Sojka, P. (2010), "Software framework for topic modelling with large corpora", In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Schindler, M. and Eppler, M.J. (2003), "Harvesting project knowledge: a review of project learning methods and success factors", *International Journal of Project Management*, Vol. 21 No. 3, pp. 219–228.

Steinbach, M. George K. and Vipin K. (2000), "A comparison of document clustering techniques", *KDD workshop on text mining*. Vol. 400. No. 1.

Wu, X., Zhu, X., Wu, G.Q. and Ding, W. (2014), "Data mining with big data", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 1, pp. 97–107

Yin, R. K. (2013), *Case study research: Design and methods*, SAGE Publications, London.

https://doi.org/10.1017/dsi.2019.266 Published online by Cambridge University Press