MARTIN, K., LIRET, A., WIRATUNGA, N., OWUSU, G. and KERN, M. 2018. Risk information recommendation for engineering workers. In Bramer, M. and Petridis, M. (eds.) Artificial intelligence XXXV: proceedings of the 38th British Computer Society's Specialist Group on Artificial Intelligence (SGAI) International conference on innovative techniques and applications of artificial intelligence (AI-2018), 11-13 December 2018, Cambridge, UK. Lecture notes in computer science, 11311. Cham: Springer [online], pages 311-325. Available from: https://doi.org/10.1007/978-3-030-04191-5_27

Risk information recommendation for engineering workers.

MARTIN, K., LIRET, A., WIRATUNGA, N., OWUSU, G. and KERN, M.

2018

Risk Information Recommendation for Engineering Workers

Kyle Martin[1 [0000−0003−0941−3111]] ✉, Anne Liret[2 [0000−0003−0620−7240]], and
Nirmalie Wiratunga[1 [0000−0003−4040−2496]], Gilbert Owusu[3], Mathias Kern[3]

[1] Robert Gordon University, Aberdeen, Scotland
{k.martin, n.wiratunga}@rgu.ac.uk
[2] BT France, Paris, France
anne.liret@bt.com
[3] British Telecommunications, United Kingdom
{gilbert.owusu, mathias.kern}@bt.com

Abstract. Within any sufficiently expertise-reliant and work-driven domain there is a requirement to understand the similarities between specific work tasks. Though mechanisms to develop similarity models for these areas do exist, in practice they have been criticised within various domains by experts who feel that the output is not indicative of their viewpoint. In field service provision for telecommunication organisations, it can be particularly challenging to understand task similarity from the perspective of an expert engineer. With that in mind, this paper demonstrates a similarity model developed from text recorded by engineer's themselves to develop a metric directly indicative of expert opinion. We evaluate several methods of learning text representations on a classification task developed from engineers' notes. Furthermore, we introduce a means to make use of the complex and multi-faceted aspect of the notes to recommend additional information to support engineers in the field.

Keywords: Case-Based Reasoning · Information Retrieval · Machine Learning · Metric Learning · Similarity Modeling · Deep Metric Learning

1 Introduction

Within any expertise-reliant and work-driven domain there is a requirement to understand the similarities between specific work tasks. Sufficiently understanding the similarity between work elements in these sectors presents an opportunity to improve transfer of experiential content [5] and provide services such as work recommendation [12]. We notice this problem frequently within field provision of services for telecommunication organisations.

Though we had previously published our findings regarding a fuzzy logic-based recommender system [12], engineers have demonstrated a reluctance in uptake of the system. In particular, it was highlighted that the system was opaque in terms of its decision-making process and that the similarity model it generated was not representative of engineer's own expert perspective. In answer to this, we have built a Case-Based Reasoning (CBR) system with greater explanatory capabilities and drawing on text documents recorded by the engineers' themselves to inform its similarity model. We have discussed the explanatory capabilities in other work [10] and thus in this paper will focus on how we utilised the complex information source of engineer notes to develop a similarity model which can act as a basis for additional information recommendation.

This work is motivated by the need to learn similarity from a user's perspective. We believe that using notes written by engineers themselves as the information source for a similarity metric will ensure that the cases retrieved through similarity-based return are more representative of this point of view. In essence, we wish to achieve a similarity model which is indicative of what a domain expert's own experiences have lead them to believe is the truth. We believe this can be achieved by basing it on experts' notations regarding the subject.

Beyond this however, the notes offer potential as a multi-faceted source which can inform a number of decision support systems. The notes are a large semi-structured source of information detailing specific experiences of human experts in the field. Thus, we view this as an opportunity to develop a corporate memory of human experience, improving the effectiveness of engineers in the field and enabling business robustness to the departure of experts from employment. Though in this work we focus on the development of the similarity model and the recommendation of information to counter explicit risks, we suggest this is an equally suitable method to achieve goals such as autonomous fault diagnosis and expert assistance recommendation.

We offer several contributions in this paper. We (1) evaluate methods of developing representations from expert-written documents for similarity-based return on the basis of their accuracy on a simple classification task. We (2) showcase a generalisable method to recommend additional information from complex and multi-faceted information sources that the notes represent. Lastly, we (3) present all of our findings with a specific use case in the telecommunications engineering domain. Though presented as a field services recommender, the concept could be adapted to fit other domains.

This paper is split into the following sections. Section 2 talks about related work. Section 3 highlights aspects of the notes which lend themselves towards developing a similarity model and additional information recommendation. Section 4 contains an evaluation of the text representation learning methods and analyses their impact learning a similarity model. In Section 5 we formalise our approach to additional information recommendation and demonstrate the completed system. Finally, in Section 6 we examine the system within the context of future work and in Section 7 we offer some conclusions.

2 Related Work

This work primarily considers the learning of similarity between telecommunication tasks. In previous work a fuzzy recommender system was used to recommend tasks to engineers [12]. However, it was found that engineers resented the lack of clarity behind its recommendations. This informed our drive to develop a similarity metric which is robust to scrutiny from an expert perspective and can be used to develop an explanation. We have previously identified the explainability aspects of this work [10], and so we focus upon learning experiential content from the expert perspective in this work.

Sharing and managing expertise is a necessary aspect of knowledge management [1]. However, knowledge transfer between expert-level users is a difficult task [4]. When seeking to automatise such an interaction, it often means that a user is directed towards another expert user, rather than presented with specific information itself [5]. This is because the information sources that store experience are necessarily complex, as experiential content is difficult to elicit and therefore difficult to query effectively [17].

Due to the above reasons, human experience composes one of the more difficult areas to manage in an organisational memory [16]. We feel that the notes recorded by engineers themselves offer a bridge between explicit corporate storage and the otherwise inaccessible content of human memory.

2.1 Learning Representations from Text

The notes detailing complex task information are recorded in text, and so in this paper we consider approaches to develop a similarity model for text documents. We compare a range of techniques to improve document representation, such as distributional and distributed methods, as well as a deep metric learner. Distributional approaches utilise statistical-based methods to measure co-occurence of words. An example of this is tf-idf [14], which calculates the term-frequency/inverse-document-frequency for each term in a document to produce a sparse representation. Distributed approaches, often associated with word embeddings such as those obtained through Word2Vec [11], produce dense representations which are representative of latent or syntactical features within individual words. These are obtained by examining the context in which words appear. For example, Word2Vec [11] uses a sliding window which considers the words both before and after the target term to provide input to a one-layer neural network. The network is exposed to multiple sliding windows containing the target term to build an understanding of its latent features based upon the context in which it appears. These latent features are represented by real numbers and can be used to build a vectorial representation of individual words. A document is then represented by an average of the word vectors (or embeddings) which it contains. This process is known as Doc2Vec [7].

Deep metric learners are a group of architectures (including the Siamese Neural Network [2, 3] and Triplet Network [15]) that use neural network techniques to learn a metric. They receive multiple examples as input simultaneously to develop embeddings which are optimised based on an objective. This objective is defined by a 'matching criteria' - a principle which identifies whether two examples are similar or not. Importantly, this matching criteria is not necessarily reliant upon class knowledge [6]. Deep metric learners develop a space which is optimised for similarity-based return by minimising the distance between examples which adhere to the matching criteria and maximising the distance between examples that do not. Due to this they have demonstrated application in areas where fine-grained similarity knowledge is important, such as face verification [15] and similar text retrieval [13].

2.2 Provision of Additional Information

Finally, since our desire is to use the generated model as the basis for recommendation of additional information, we review methods of similarity-based recommendation. From the literature, we can observe a number of examples of similarity-based recommendation being used to take advantage of complex sources and provide decision support services for experts. One such field is interpersonal trust modeling [8, 9].

In [8] and [9], the authors utilise similarity measures to develop a prediction of how effectively an expert will integrate with a team to complete group-work reliant tasks. This is based on understanding of interpersonal trust between group members gained from individual member feedback. A similarity model is then generated using the similarity between members' prior task preferences, as gained from ratings that each member of the group had given to tasks they themselves had previously completed. This is similar in nature to the approach we adopt in this paper for additional information recommendation. We are reliant on users to explicitly describe reasons for task failure to form the basis of our system to identify risk information. However, this information is captured by the business anyway, and so does not require additional data collection from the engineer as a rating would provide. Furthermore, the information we are able to extract from the text documentation is more complex than a single numerical rating value, and therefore more valuable.

3 Similarity from an Expert's Perspective

It is common within complex services provisioning that the personnel which fulfill the required technical work gradually become highly skilled in their domain. In this paper specifically, we highlight the telecommunications engineering force whom develop expertise in network equipment installation and repair. To ensure continuous service delivery, they traditionally are allocated tasks. A task, in this scenario, represents either a time-constrained action to perform on a piece of equipment or an investigation regarding a delayed step of a wider network process. Field engineers record information about the tasks they have completed in text documents called ”notes”. These notes form an heterogeneous base of mixed types of information, such as work order, identified problem, failure reason, task progression, task context, and sometimes informal recommendation. The notes are categorised based upon their contents. For example, Order notes contain information about the work requirements of a task. Details of task completion are stored in Closure notes, while records of task failure and the reason behind it are stored in Further notes. Lastly, an engineer can enter additional miscellaneous information about a task in its in User notes.

The notes are a semi-structured source of expert information describing a specific task. However, they are made complex by the fact that different only certain note types may be present in certain tasks. For example, a task which has never been attempted will only be associated with Order notes, while there may be both Further and Closure notes describing another task if it had been failed at least once before it was successfully completed. Furthermore, a task will often not only be associated with multiple different types of notes, but also multiples of the same note type (i.e. two Further notes if the task has been failed twice). However, certain note types (such as Order notes) will never be duplicated. Figure 1 displays the relationship between notes and tasks for two examples.

In this paper, we demonstrate a similarity model generated based upon Order notes, which are representative of the original work order calling for the task to be completed.

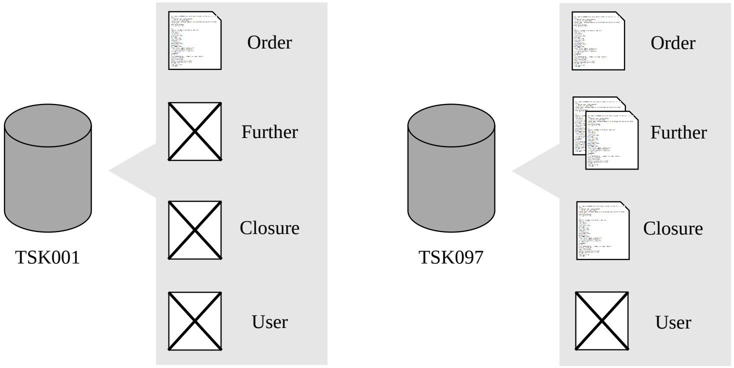


Fig. 1: The relationship between tasks and notes. Task TSK001 is an example of a newly created task, where only the Order notes have been generated. Meanwhile, TSK097 is an example of a task that has been attempted twice and failed (notice two Further notes) before being successfully completed (detailed in the Closure notes). In both instances the engineer has declined to add more information in the User notes.

Since an Order note is created at the same time as the original task, this guarantees that a task can be used to query the model immediately upon creation. This is an important component of a timely system, as it would be less useful to query the model after a task has already been allocated, or even failed.

We also propose a method to use the notes for recommendation of additional information. For this purpose, we consider the specific use case of risk information recommendation and so we are primarily interested in the reasons that govern task failure. These are recorded in Further notes, while details of successful task completion are recorded in Closure notes. However, as aforementioned, neither Further nor Closure notes are available for every task, as these note types are only generated when a task is attempted and failed or succeeded respectively. This has two consequences: (1) that a task which is completed on first attempt (the most common scenario) does not become associated with any point of failure and so is not associated with a risk and (2) that newly generated tasks will not be associated with either of these note types.

Since it is necessary for the similarity model to be equally valid across the entire case-base, these note types would be ineffective to use as a basis for a similarity model. Hence we utilise a similarity model which is generated based upon a shared point of commonality across all of the case-base (Order notes). However, Order notes alone are not sufficient to develop a recommendation of additional information. To provide this, we need the information that is present within these specialised note types. The result is that we will only consider a specific subset of tasks in which these notes are present (i.e. Further notes detailing task failure) for return. We are therefore considering multiple aspects of the notes in this system - we develop a metric as based upon a point of commonality (Order notes) and can develop a recommendation based upon specific relevant task notes (Further notes).

4 Developing a Similarity Model from Text

In this work we compare several methods of learning representations for text documents on the basis of their performance on a simple similarity-based return classification task. This allows us to determine the most suitable method to use as a basis for our model. Specifically, we consider a distributional representative (tf-idf), a distributed representative (doc2vec) and a deep metric learner (Siamese Neural Network (SNN)). In addition, as research has demonstrated that the performance of a deep metric learner can be enhanced through the selection of a suitable training strategy [15], we also consider an SNN which uses DYNEE sample selection.

Term frequency-inverse document frequency (tf-idf) is a statistical measure that calculates a real value for each term in a document. The value for each term is calculated by dividing the frequency of its usage within a document over the number of documents which contain the term within the corpus [14]. Therefore, each feature of a document vector is a value which represents an individual word from the corpus vocabulary and so vectors can be very sparse. As a result, tf-idf becomes steadily less effective in large corpora with a varied vocabulary.

Document-2-Vector (Doc2Vec) [7] is an extension of the Word2Vec algorithm [11]. Word2Vec uses contextual knowledge from the measurement of word co-occurrence to capture latent or syntactical features and develop word embeddings. The result is a feature space where words that have similar contexts exist close together. To retrieve a representation for a document (Doc2Vec), the embeddings for terms within the document are simply averaged. Sparseness of representation is thereby avoided, as feature values are not reliant on specific words being present.

The Siamese Neural Network (SNN) is a deep metric learner. It has a unique neural architecture comprised of two matching sub-networks with identical weights and parameters. This allows the network to receive example input in pairs, thereby enabling the network to train based on the relationships that exist between examples. When input to the network, pairs of examples are labeled on the basis of a 'matching criteria'. The metric space which is learned by the SNN is optimised to fit the stated matching criteria, such that positive pairs (in which member examples adhere to the matching criteria) exist close together, while negative pairs (in which members do not) exist far apart.

Learning of representations in this manner is controlled by SNN's unique contrastive loss function, introduced in [3] and we summarise here. Contrastive loss is split into two elements; L_G for positive (genuine) pairs and L_I for negative (impostor) pairs. The use of both genuine and impostor error means that the similarity metric can be directly learned by the network through the comparison of the actual pair label Y_A (equal to 0 for genuine and 1 for impostor pairs) and the distance between pair members, D_W. This means that distance between constituents of genuine pairs are minimised over the course of training, whilst ensuring that impostor pairs maintain at least a set margin of M distance apart.

$$L_G = (1 - Y_A) \cdot {D_W}^2 \tag{1}$$

$$L_I = Y_A \cdot (max(0,\ M - D_W))^2 \tag{2}$$

$$L = L_G + L_I \tag{3}$$

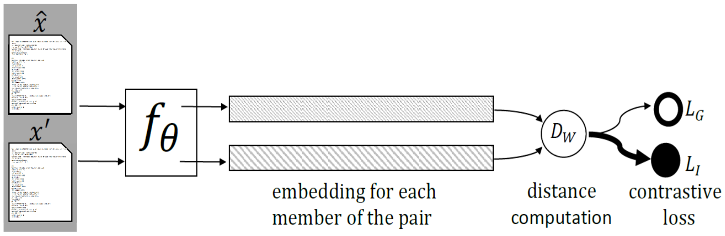


Fig. 2: The Siamese Neural Network (SNN) architecture

The output of the identical sub-networks form feature embeddings for each member of the input pair. During training it is these embeddings that are used for any distance computations, thereby ensuring iterative model refinement. An example SNN architecture is presented in Figure 2.

As SNNs receive pairs of examples as input, this presents an additional parameter that can be fine-tuned. In its basic form, the network will receive randomly generated pairs as input during training. However, research has demonstrated that a higher optima and quicker convergence can be achieved if an appropriate pairing strategy is adopted. With this in mind, we also considered an SNN supported by DYNEE sample selection. DYNEE is a pairing strategy which optimises pair creation through the use of a combined exploration and exploitation strategy. DYNEE exploits knowledge gained in previous stages of training by identifying pairs which had generated high loss values. It then uses them to augment a randomly generated pair set which provides exploratory coverage of the space. This is repeated over the course of training to ensure that the pair set is formed from contemporary information.

4.1 Evaluation

In this section we evaluate the vectorial representation of notes gained through each method - tf-idf, Doc2Vec, SNN and an SNN with DYNEE sample selection - to determine which develops the best representation in terms of accuracy on a classification task. Although a subject study would be desirable to understand whether the model develops a score which is representative in an expert's opinion, we will empirically evaluate the model using a simple classification task as a proxy. In future work we plan to complete such a subject study to accompany and strengthen this empirical evaluation.

4.2 Experimental Setup

For the purposes of comparison we have created a simple classification task where notes are classified according to one of four work types. The goodness of each of the learned representations are assessed by their performance on this classification task.

We extracted two months of Order notes written by telecommunication engineers between March and April 2018. We filtered out any note which contained less than 50 characters, as we judged them not to be adequately meaningful. This resulted in a dataset of 1,610 notes split into four classes - Cabling (CAB - 227 notes), Jointing (JRT - 789 notes), Overhead (OVH - 503 notes) and Power Testing (PTO - 91 notes). These classes represent the primary required competence which is associated with each note.

The dataset was split into train and test and evaluated using 5-fold cross evaluation. Each of the abovementioned methods were used to generate a representation for the notes [1]. We then used k-nearest neighbour (with a k value of 3) for similarity-based return. The Doc2Vec feature size was 300 as this was found to be the optimal parameter through empirical evaluation. For the SNN implementation using DynEE, pair selection was repeated every 5 epochs. The α exploitation ratio used for DynEE was $^{|P|}/_{10}$. This means that 10% of the training pairs at any one time were formed through exploitation, while the remainder were randomly generated to provide exploration.

4.3 Results

The results are presented in Table 1[2]. It is unsurprising that deep metric learners perform better than either tf-idf or Doc2Vec on the classification task, as they are designed to produce embeddings which are optimal for similarity-based return. We can observe that the SNN with the DynEE training strategy achieved the highest accuracy, though this was only marginally greater than its vanilla counterpart. This is likely due to the small size of the dataset. Tf-idf performed the worst. This was as expected, because as a statistical measure tf-idf does not consider the context of terms.

Table 1: Results of representation learning methods on a classification task.

Architecture	Accuracy (%)
Tf-Idf	62.24
Doc2Vec	63.79
SNN	66.25
SNN DynEE	66.83

Figure 3 presents a principle component analysis of the representation distribution gained from the Doc2Vec and SNN on a multi-dimension scaling scatter plot. We can observe that the representation distribution gained from SNN demonstrates significantly better similarity clusters. It confirms that concepts learned by SNNs form better clusters around class boundaries when compared to Doc2Vec. This also provides supporting evidence towards the performance gains observed with SNNs.

[1] Both SNN sub-network architectures were comprised of 3-layer perceptrons which used an SGD optimizer, ReLU activations and were trained for 250 epochs.

[2] The displayed results for SNNs used Doc2Vec embeddings as input, as they performed better. When using tf-idf representations as input, both SNNs still outperformed tf-idf and Doc2Vec.

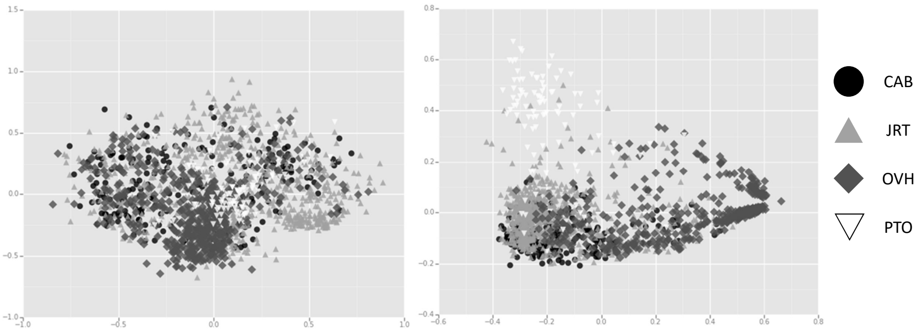


Fig. 3: Representation distribution gained from Doc2Vec (left) and SNN (right)

5 Recommendation of Additional Information

Additional information can take many forms and assist an engineer in the field in different ways. We formally view additional information in this paper as the provision of extra knowledge which can contribute to the successful completion of a task. Of particular interest is additional information which may allow a user to pre-emptively identify possible task failure and potentially avoid it. In this manner, we hope to either prevent task failure or 'fail fast' such that minimum resources are wasted on a doomed task.

We propose a method to make further use of the engineering notes. We suggest that based upon the developed similarity model, we can recommend additional information to users with the purpose of supporting their work. Specifically, we will identify the likelihood of potential risk categories to an incoming task and make a recommendation based on knowledge from the notes to counter the risk where appropriate. Though we focus on recommendation of risk information in this work, the same principle can apply to other problems and domains with similarly recorded expertise.

5.1 Formalising Risk

Firstly, we extract a set of risk categories from the Further notes of previously failed tasks. These categories are an abstraction of specific risks that are collected into a single related concept (i.e. both ”the customer was not ready” and ”the customer was not present” would fall into the Customer risk category). This presented us with six risk categories - Contractor, Customer, Duct Blockage, External Event, Planning and Time. These categories were formed based upon feedback from telecommunication engineering experts and the most common sources of risk. For example, though Duct Blockage is a reasonably specific point of failure, it is a very common one. Equally, while the category External Events covers many different hazards (i.e. dangerous animals, adverse weather, etc), it is much rarer for any individual risk to cause task failure.

We then label each failed task based upon the risk category which caused its failure. Note that any given task may have been failed more than once and so can be associated with multiple labels. Also note that these labels only apply to tasks which have already

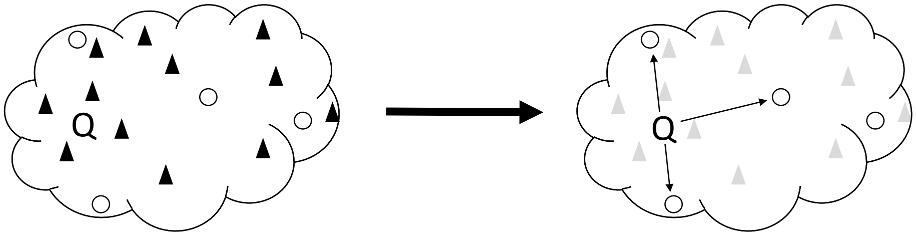


Fig. 4: Vote weighted by similarity. If the triangles represent successfully completed tasks, while the circles represent tasks that failed at least once, then we can observe only the latter feed into the weighted by similarity vote.

been failed - we do not generate risk category labels for tasks that were successful on their first attempt or have yet to be attempted (i.e. the tasks that lack Further notes).

To perform the recommendation of risk information, we can then submit a query case to the similarity model to retrieve a return set. However, instead of considering all possible tasks, we only consider tasks which are associated with Further notes (and therefore at least one risk label). We can then perform a vote weighted by similarity to gauge the likelihood of a given risk occurring in the query task (see Figure 4). We perform this vote in the following manner.

Let us describe an individual risk category as r and a function to induce a score for the risk category of an unseen example $r()$. We will also describe a task as x and our full set of task examples as X. Similarly we will identify a failed task as $\hat{x}$, such that $\hat{x} \in \hat{X}$ and $\hat{X} \subset X$. We can retrieve a label for a given failed task using the function $y(\hat{x})$. We compare a query, q, with its set of nearest failed neighbours, $\hat{x}^{NN}$. To develop a score, each risk category is calculated by the following formula:

$$r(q) = \frac{1}{k} \sum_{i}^{k} sim(x_i^{NN}, q) \cdot s_r \tag{4}$$

where k denotes the neighbourhood size parameter and s_i denotes a binary value 'switch' which is set to 1 if x_i^{NN} has previously failed due to the given risk (i.e. $r = y(\hat{x})$), or 0 otherwise. What this means is that a risk category's score is based upon a similarity weighted vote of its nearest failed neighbours. In this manner, we can develop a score of the likelihood for the occurrence of each risk in a given query.

5.2 Generating Recommendations

Though it is useful to demonstrate the likelihood of individual risks to an engineer, this is not necessarily helpful if they do not understand how to circumvent those risks. Therefore, we also generate a recommendation to answer any sufficiently likely risks. This is achieved by comparing each individual risk category score against a threshold. We have three possible classifications of risk - Low ($r(q) < 30\%$), Medium

($r(q) > 30\%$ and $r(q) < 60\%$) and High ($r(q) > 70\%$). The recommendations themselves are based upon the most common solution successful solution derived from the Closure notes of previously failed task. For example, most of the Closure notes suggest that many task failures relating to the Customer label can be avoided by phoning the customer ahead of time. It is worth noting that this will not necessarily prevent the task itself from failing. If the Customer has still not completed necessary pre-work, then the task will fail regardless. It does however offer an opportunity to 'fail fast' (i.e. prevent the engineer wasting time traveling to the customer's location). This in itself will improve productivity, as the engineer will then be free to complete another task.

Although simple, these prototypical risk solutions are easily generalisable within any of the given risk categories. Furthermore, as they are representative of the notes, they draw on what the experts themselves (in this case telecommunication engineers) have commonly found to be a successful approach. Though the recommendations are currently confined to a singular generalisable recommendation for any given risk category, in future work we seek to develop personalised representations based upon the notes associated with the nearest failed neighbours which have since been succeeded based upon Closure notes.

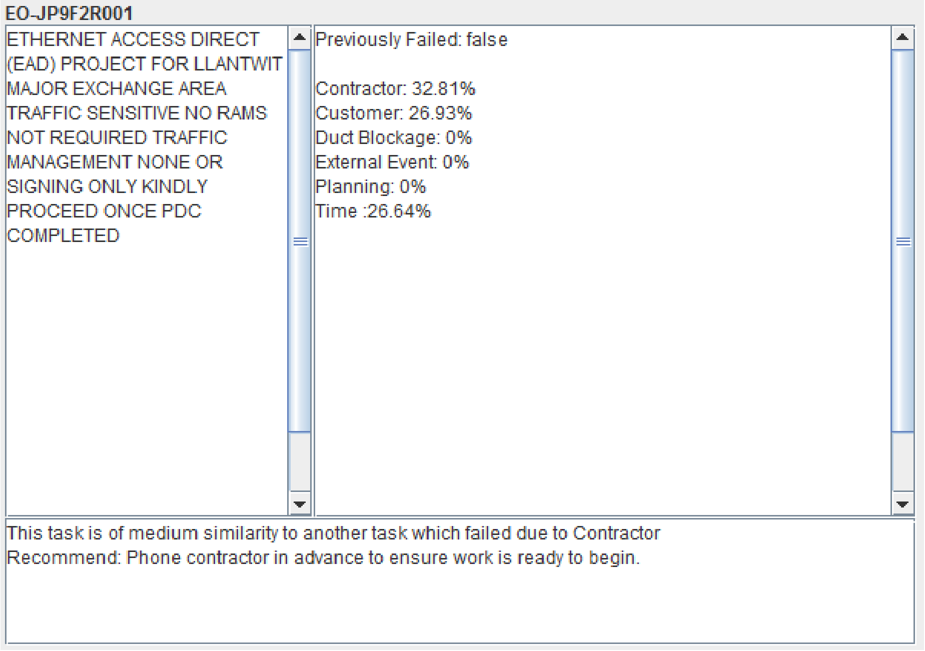
An example of the system is presented in Figure 5. Note the text area on the left provides details of the original task, while the window on the right details the scoring across the list of risk categories. The bottom window is used to provide a recommendation - notice that the greatest risk being Contractor is highlighted and the system recommends that the user contact the Contractor in advance to ensure that the work is ready to begin.

5.3 Previously Failed Tasks and Progressed Tasks

Field service provisioning is a field where large scale jobs can frequently occur. These jobs are usually broken down into a series of related, often sequential, tasks which we describe as a work chain. Therefore, this environment must be considered when recommending additional risk information. Equally, there is potential for tasks to have been failed by one engineer before being attempted by another. Thus it is necessary for our system to consider at least some evidence of task history in order to make its recommendations.

We adopt a strict stance towards failure in task history for our additional information recommender; if a task, or any of the previous tasks in its work chain, has ever been failed previously, then we make a strong recommendation to counter this risk. The system will still display scoring for other risk categories, but will highly recommend that action be taken to answer this specific risk. Furthermore, it will highlight that this task (or a member of its work chain) has been failed in the past and provide the Further notes regarding the failure. These notes include contact information for the engineer that previously attempted the task, as well as specific information on the failure. This enables the engineer to form a response to the risk or contact the engineer for further information as required.

An example of the recommendation made regarding a previously failed task is shown in Figure 6. We can observe that the right text area has been extended to include the Further notes of the previously failure. Note also that the bottom text area


EO-JP9F2R001

ETHERNET ACCESS DIRECT (EAD) PROJECT FOR LLANTWIT MAJOR EXCHANGE AREA TRAFFIC SENSITIVE NO RAMS NOT REQUIRED TRAFFIC MANAGEMENT NONE OR SIGNING ONLY KINDLY PROCEED ONCE PDC COMPLETED

Previously Failed: false

Contractor: 32.81%
Customer: 26.93%
Duct Blockage: 0%
External Event: 0%
Planning: 0%
Time :26.64%

This task is of medium similarity to another task which failed due to Contractor
Recommend: Phone contractor in advance to ensure work is ready to begin.

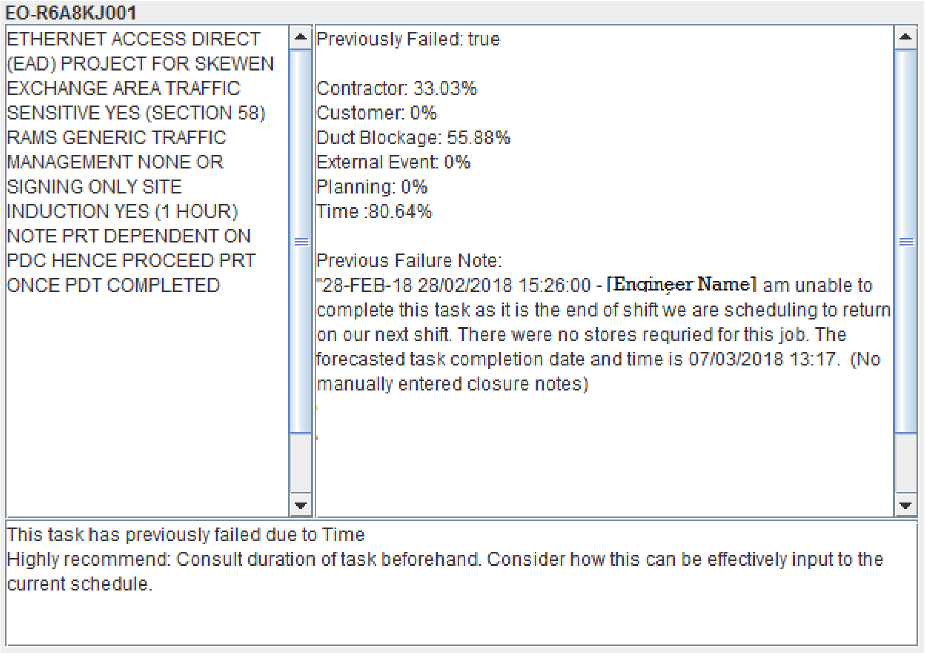
Fig. 5: Additional information recommendation for a medium risk task

highlights which risk category caused this task to fail on its previous attempt. If more than one Further note was associated with this task, then all previously written Further notes would appear in the right text area. Similarly, if a previous task in the work chain had failed, the bottom window would identify this.

6 Future Work

In future work we would like to evaluate the risk recommendation method demonstrated in this paper by performing a full subject study on engineers to retrieve qualitative feedback on the system. This would improve the robustness of the empirical evaluation to measure embedding goodness.

Ultimately, we would like to offer explanations of why a certain risk has been identified and offer recommendations on how best to circumvent the identified risk which are both personalised to the given task. We believe these personalisation can also be produced from the engineers' notes. In our current work, we are attempting to learn a generalisable breakdown of the notes, such that we can quickly identify the most appropriate passage to present to an engineer by way of an explanation. A similar method could also be applied to find the most likely solution to a risk. We believe this could be a very useful tool in the field to support engineering workers - a system which is able to pinpoint similar tasks as a method to provide additional information, support this infor-



EO-R6A8KJ001

ETHERNET ACCESS DIRECT (EAD) PROJECT FOR SKEWEN EXCHANGE AREA TRAFFIC SENSITIVE YES (SECTION 58) RAMS GENERIC TRAFFIC MANAGEMENT NONE OR SIGNING ONLY SITE INDUCTION YES (1 HOUR) NOTE PRT DEPENDENT ON PDC HENCE PROCEED PRT ONCE PDT COMPLETED

Previously Failed: true

Contractor: 33.03%
Customer: 0%
Duct Blockage: 55.88%
External Event: 0%
Planning: 0%
Time :80.64%

Previous Failure Note:
"28-FEB-18 28/02/2018 15:26:00 - [Engineer Name] am unable to complete this task as it is the end of shift we are scheduling to return on our next shift. There were no stores requried for this job. The forecasted task completion date and time is 07/03/2018 13:17. (No manually entered closure notes)

This task has previously failed due to Time
Highly recommend: Consult duration of task beforehand. Consider how this can be effectively input to the current schedule.

Fig. 6: Additional information recommendation for a previously failed task

mation through an explanation and offer a recommendation on how to take advantage of that knowledge effectively.

7 Conclusion

In conclusion we have demonstrated a similarity model which is built upon textual documents written by experts as a source. Furthermore, we have advanced one step beyond that to use our similarity model to generate additional information to support experts while in the field. We have focused on the pre-emptive identification of risk categories and build on common solutions from the notes to recommend a possible work around for these risks.

References

1. Ackerman, M.S., Pipek, V., Wulf, V.: Sharing expertise: Beyond knowledge management. MIT press (2003)
2. Bromley, J., Guyon, I., LeCun, Y.: Signature verification using a 'siamese' time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7**(4), 669 – 688 (August 1993)
3. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proc. of the Conf. on Computer Vision and Patter Recognition. pp. 539 – 546. CVPR '05, IEEE Computer Society, Washington, DC, USA (June 2005)

4. Dörner, C., Pipek, V., Won, M.: Supporting expertise awareness: Finding out what others know. In: Proceedings of the 2007 Symposium on Computer Human Interaction for the Management of Information Technology. CHIMIT '07, ACM (2007)
5. Göker, M.H., Thompson, C., Arajärvi, S., Hua, K.: The pwc connection machine: An adaptive expertise provider. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) Advances in Case-Based Reasoning. pp. 549 – 563. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
6. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Deep Learning Workshop. ICML '15 (July 2015)
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML'14, JMLR.org (2014)
8. Malinowski, J., Keim, T., Weitzel, T., Wendt, O.: Decision support for team building: incorporating trust into a recommender-based approach. In: The Ninth Pacific Asia Conference on Information Systems. pp. 604 – 617 (2005)
9. Malinowski, J., Weitzel, T., Keim, T.: Decision support for team staffing: An automated relational recommendation approach. Decision Support Systems **45**(3), 429 – 447 (2008), special Issue Clusters
10. Martin, K., Liret, A., Wirtaunga, N., Owusu, G., Kern, M.: Explainability through transparency and user control: A case-based recommender for engineering workers. In: XCBR Workshop. ICCBR 2018 (July 2018)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
12. Mohamed, A., Bilgin, A., Liret, A., Owusu, G.: Fuzzy logic based personalized task recommendation system for field services. In: Bramer M., Petridis M. (eds) Artificial Intelligence XXXIV. SGAI 2017.Lecture Notes in Computer Science, vol 10630. pp. 300–312. Springer, Cham, Cambridge, UK (December 2017)
13. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In: Rep4NLP@ACL (2016)
14. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. pp. 133 – 142 (2003)
15. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 815 – 823. CVPR '15, IEEE Computer Society, Washington, DC, USA (June 2015). https://doi.org/doi:10.1109/cvpr.2015.7298682
16. Walsh, J.P., Ungson, G.R.: Organizational memory. The Academy of Management Review **16**(1), 57–91 (1991)
17. Wright, G., Ayton, P.: Eliciting and modelling expert knowledge. Decision Support Systems **3**(1), 13–26 (1987)