# Improving spare part search for maintenance services using topic modelling

**Document status and date:**
Published: 01/01/2020

**Document Version:**
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Download date: 05. Oct. 2023

# Improving Spare Part Search for Maintenance Services using Topic Modelling

Anastasiia Grishina[a], Milosh Stolikj[b], Qi Gao[b] and Milan Petkovic[a,b]

[a]*Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, The Netherlands*
[b]*Philips Research, High Tech Campus 34, 5656 AE, Eindhoven, The Netherlands*

## Abstract

To support the decision-making process in various industrial applications, many companies use knowledge management and Information Retrieval (IR). In an industrial setting, knowledge is extracted from data that is often stored in a semi-structured or unstructured format. As a result, Natural Language Processing (NLP) methods have been applied to a number of IR steps. In this work, we explore how NLP and particularly topic modelling can be used to improve the relevance of spare part retrieval in the context of maintenance services. A proposed methodology extracts topics from short maintenance service reports that also include part replacement data. An intuition behind the proposed methodology is that every topic should represent a specific root cause. Experimental were conducted for an ad-hoc retrieval system of service case descriptions and spare parts. The results have shown that our modification improves a baseline system thus boosting the performance of maintenance service solution recommendation.

## Keywords

Entity retrieval, spare part search, decision support, maintenance services, natural language processing, topic modelling

## 1. Introduction

Information retrieval systems are gaining importance in various industrial applications. We can observe the emergence of knowledge-based systems that support the decision-making process in construction, aviation, equipment maintenance and other areas [1, 2]. In these settings, knowledge is frequently extracted from data that is captured in legacy systems using natural language and stored in a semi-structured or unstructured format. As a result, linguistic and statistical NLP methods have been applied to a number of IR steps, such as document and query modelling, query expansion and search result clustering based on semantic similarities [3, 4, 5, 6].

In this work, we explore how NLP and particularly topic modelling can be used to improve spare part retrieval that serves the purpose of medical equipment maintenance. In particular, we focus on remote system diagnostics that takes place when the equipment malfunctions, i.e. stops working according to its specification. The problem may be resolved in several ways, one of which is the replacement of one or more (malfunctioning) parts. We conducted our research in the context of an ad-hoc entity retrieval

system which helps engineers to search for relevant historical service reports and identify the most probable service solution. Therefore, target retrieval entities are equipment components, i.e. parts to be replaced. In practice, one case may require multiple parts to be replaced.

To address the challenge of spare part retrieval, we create an NLP pipeline that pre-processes short textual descriptions of maintenance activities and apply topic modelling to categorize the descriptions of past cases. From relevant maintenance service reports, the proposed methodology extracts topics each of which may indicate a specific root cause. Once categorized, cases and parts would be easier to examine and more relevant to a particular type of failure. An engineer can address topics seuqentially and choose among parts related to the same topic. Therefore, we exploit term co-occurrences and their semantic correspondences using topic modelling to enhance the relevance of target entities retrieval. Although the use case assumes that a number of parts will be ultimately suggested based on past maintenance records, the problem statement does not fall under the vastly explored area of recommender systems that involves user preference modelling.

To evaluate the difference introduced by the proposed component, we use IR metrics that are customized to characterize the relevance and completeness of a set of retrieved entities. They measure how far in the list of search results all the required parts are present, indicate if at least one required entity is retrieved and whether all needed

parts are present among top K search results.

The main contributions of the work are as follows:

- we enhance the performance of an industrial entity retrieval system by learning semantic correspondences between short historical descriptions of events associated with the entities;

- we approach the challenge of spare parts retrieval in remote system diagnostics and maintenance of industrial equipment using topic modelling to group extracted historical cases and parts under topics that should represent failure root causes;

- we evaluate the proposed method on a real world dataset using customized information retrieval metrics.

The remainder of this paper is organized as follows. We present the problem formulation and a baseline part retrieval system in Section 2. The methodology of combining the text mining pipeline and the entity retrieval process is described in Section 3. Section 4 is dedicated to a dataset description and methods implementation. We discuss experimental results in Section 5 and related work in Section 6. The paper is concluded by Section 7 where we also mention possible directions for future work.

## 2. Problem Description

In the scope of this work, entity descriptions are composed of equipment characteristics and represented by maintenance case reports registered in the retrieval system. Entities to be retrieved are the parts recommended for replacement to troubleshoot a machine referred to in a new malfunction report. Queries may contain various characteristics of a new maintenance case that should be treated by a maintenance service team. An entity, i.e. a spare part, is identified with a unique ID and is related to a case description. One historical maintenance case can have several parts associated with it, similarly, a new service case may require a set of different parts.

The knowledge base of maintenance cases is updated with the help of service engineers. They submit maintenance reports for every equipment failure or customer complaint as short technical texts often in multiple languages (English and a locally spoken language). Each historical report includes a number of logs such as time of customer complaint registration, a textual description of maintenance

activities and IDs of parts used to solve the issue. Hence, the reports might contain abbreviations, software logs sent by a machine as well as natural language descriptions of a machine state on every step of the maintenance process. Closed cases are uploaded to the collection of historical cases that could be mined using the above mentioned ER system.

To present the setting in a formal way, let $q$ be a query performed by a service engineer while working on a case. We will use the term *query case* to indicate such cases. Each query is associated with a single maintenance case. The list of parts replaced in a case $c$ is $P(c)$. We use $C(q)$ to denote a list of cases retrieved for the query $q$. A set of parts replaced in all retrieved cases is denoted by $P(q) = \cup_{c \in C(q)} P(c)$, and a set of ranked parts recommended for replacement is expressed by $P_R(q) \subseteq P(q)$.

## 3. Methodology

The method proposed in this work combines a baseline entity retrieval setting and an add-on topic modelling component as described below.

### 3.1. Baseline Entity Retrieval System

The baseline entity search system in question is empowered with a two-step retrieval mechanism. A database of entity descriptions lies in the foundation of the mechanism. It consists of entity descriptions retrieval followed by the final entity retrieval and ranking as explained in detail below.

#### 3.1.1. Retrieval of Entity Descriptions

At the first step of the entity search, a system retrieves relevant descriptions using a Vector Space Model (VSM) with Okapi BM25 similarity score [7, 8]. VSM is a document and query representation model that converts texts to N-dimensional vectors of term weights, where N is the number of words in a dictionary. Terms are simply the words or groups of words present in the collection of documents. The dictionary is built from a text corpus and includes distinct terms. The intuition behind VSM is that retrieved documents will be ranked according to a similarity function computed for a query and a document, i.e. vectors in a vector space.

In the context of our problem description, for a query $q$ containing keywords $\{q_i\}_{i=1}^{n}$ and a maintenance case description $c$ with fields $\{c_j\}_{j=1}^{m}$, Okapi

BM25 similarity score could be expressed as follows:

$$BM25(q, c) =$$
$$= \sum_{j=1}^{m} \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, c_j) \cdot (k_1 + 1)}{f(q_i, c_j) + k_1 \cdot \left(1 - b + b \cdot \frac{L_{c_j}}{L_{avg_j}}\right)}.$$

$$(1)$$

Here, $f(q_i, c_j)$ is the frequency of the keyword $q_i$ in a field $c_j$ of the case description $c$. $L_{c_j}$ is the length of the field $c_j$ in terms of words, and $L_{avg_j}$ is the average length of the field $j$ in descriptions of all cases in the collection $C$. Variables $k_1$ and $b$ are tuning parameters that control how much every new occurrence of a term impacts the score and the document length scaling correspondingly. Inverse Document Frequency is calculated as:

$$IDF(q_i) = \log\left(\frac{M - n(q_i) + 0.5}{n(q_i) + 0.5}\right), \quad (2)$$

where $M$ is the total number of cases, i.e. $M = |C(q)|$, and $n(q_i)$ is the number of case descriptions that contain the query term $q_i$. Therefore, the case $c^{i_1} \in C(q)$ is ranked higher than $c^{i_2} \in C(q)$ iff $BM25(q, c^{i_1}) > BM25(q, c^{i_2})$.

### 3.1.2. Entity Retrieval and Ranking

The second step realizes the entity retrieval. It ranks spare parts associated with the retrieved cases based on the frequency of their occurrence and the rank of the case where they occur. Thus, the most frequent parts that occur in top ranked cases appear higher on the final list of retrieved parts than a part that appears the same number of times lower on the case list. Several proprietary filters are applied as well, but they do not affect the methodology. The algorithm for part recommendation is presented in Algorithm 1.

### 3.2. Topic Modelling Component

Transformation of the historical cases and parts retrieval pipeline is performed by adding a component that groups retrieved cases under a number of topics and ranks the parts within the topics. Figure 1 shows the baseline architecture (a) and the modification that includes the proposed topic modelling component (b).

The topic modelling component could be considered as an individual NLP pipeline with a number of steps. The pipeline includes tokenization, lemmatization, removal of stop phrases, building a dictionary of tokens, term weighting and topic modelling. Tokenization of the text refers to splitting it into units or

---

**Algorithm 1** Part Recommendation

**Input:** Query $q$ associates with maintenance case $c$, number of parts to recommend $K$
**Output:** A list of recommended parts
  $count \leftarrow \{\}$  ▷ # occurrences of part combinations
  $P(q) \leftarrow \{\}$                    ▷ retrieved parts
  $P_R(q) \leftarrow \{\}$                  ▷ recommended parts
  **for** $c \in C(q)$ **do**:
    $P(c) \leftarrow$ **get part IDs**$(c)$
    **if** $P(c) \in P(q)$ **then**
      $count(P(c)) \leftarrow count(P(c)) + 1$
    **else**
      $P(q) \leftarrow P(q) \cup P(c)$
      $count(P(c)) \leftarrow 1$
    **end if**
    **sort**$(P(q), using=count(P(c)), order=DESC)$
    **for** $P(c) \in P(q)$ **do**
      $P(q) \leftarrow P(q) \cup P(c)$
      **drop duplicates**$(P(q))$
    **end for**
  **end for**
  $P_R(q) \leftarrow$ **top K**$(P(q))$

---

tokens that represent individual words or sometimes groups of words [9]. The process of lemmatization involves finding the initial forms of the inflected words, also referred to as root forms or lemmas. A lemma is a word in its canonical form that exists in the dictionary of the used language. For example, the lemma for *do, doing, did* is the word *do*. Next, term weighting refers to assigning weights to tokens. We utilize term frequency or bag-of-words weights as a term weighting scheme. It associates a term with a weight proportional to the frequency of the term occurrence in the corpus of documents. For topic modelling, we use Latent Dirichlet Allocation (LDA), one of the most popular algorithms for automatically extracting topics. LDA is based on the generative probabilistic language model [10]. The purpose of LDA is to learn the representation of a fixed number of topics and derive the topic distribution for every document in a collection. Every maintenance service case is assigned a topic according to the maximum probability of the case belonging to a topic.

## 4. Evaluation

In this section, we describe the real world dataset that is extracted from the baseline part retrieval system. We also discuss the metrics used to evaluate the performance of the baseline system and compare it to

**Figure 1:** Integration of the topic modelling component *(b)* in a baseline two-step document and part retrieval system *(a)*.



**Figure 2:** Distribution of queries over the number of cases and parts retrieved in response to the queries.

the configuration with the integrated topic modelling component.

## 4.1. Dataset Description

For our experiments, we use a proprietary dataset composed of historical maintenance cases. Textual fields of case descriptions have been aggregated into one field per maintenance case and serve as input to LDA during training and testing stages. The majority of cases are written in mixed languages. Figure 2 presents the distribution of the number of queries over their characteristics: the number of retrieved service cases, retrieved ranked parts to replace and parts replaced in the query case. The majority of queries retrieved up to 200 similar case descriptions, however, this number could reach 1000 cases. The number of unique recommended parts retrieved from these cases was below 350 in general, while the majority of queries retrieved 0-10 parts. The number of parts required to treat a maintenance case associated with the query was equal to 5 or less in most of the query cases.

For building the LDA model, we use a subset of historical cases written in English. The training set contains data from 101,026 different maintenance cases. For the test set, we use a sample of 1,564 queries performed by service engineers, together with the corresponding cases returned as search results: $(q, C(q))$. Cases returned for the queries may have non-empty intersection with the training dataset, however, the cases for which the queries had been created were excluded from the training set.

## 4.2. Evaluation metrics

Top $K$ ranked parts are used to estimate *success*, *completeness*, *recall* and *min_top_k* metrics. Metric@K is computed for a set of retrieved parts $|P_R(q)| \leq K$. The operator $|\cdot|$ applied to a set defines the count of set elements. The metrics are calculated as follows:

$$completeness@K(q) = \begin{cases} 1, & \text{if } P(c) \subseteq P_R(q), \\ 0, & \text{if } P(c) \nsubseteq P_R(q); \end{cases}$$

$$success@K(q) = \begin{cases} 1, & \text{if } |P(c) \cap P_R(q)| > 0, \\ 0, & \text{if } |P(c) \cap P_R(q)| = 0; \end{cases}$$

$$recall@K(q) = \frac{|P(c) \cap P_R(q)|}{|P(c)|};$$

$$min\_top\_k@K(q) = k, \quad k \leq K \text{ and}$$
$$completeness@k(q) = 1;$$

*Completeness* measures whether all the used parts were suggested for a troubleshooting report, *success* shows if any consumed part was listed among retrieved parts and *recall* indicates the ratio of retrieved parts that were consumed to the total number of consumed parts. An additional metric *min_top_k* is used to estimate how far in the list of retrieved parts one could find the full list of consumed parts in the query case and returns null if such $k$ does not exist. As a baseline, we use the initial part retrieval strategy and its statistics for the whole set of retrieved and ranked parts $P_R(q)$. Once topics are computed, the metrics are estimated for parts associated with the cases in every topic $t$, i.e. a subset of cases and, therefore, parts: $P_R(q)^{(t)} = \{P(q) \mid c \in C(q)\ \&\ c \in t\}$ instead of $P_R(q)$. We discard query cases that did not include information whether some parts were consumed or not (i.e. missing data). If a case did not require any part replacement, we utilize an artificial part called "No parts" and assign an ID to it. In this way, for query cases that were solved without part replacement it is possible to evaluate the performance of part retrieval. The top ranked part in this situation should be "No parts".

### 4.3. Implementation

The first step of the initial ER system is powered by Elasticsearch [11]. It performs indexing of the documents in the knowledge base and retrieves them according to Okapi BM25 ranking with default tuning parameters $k_1 = 1.2$ and $b = 0.75$.

For the add-on topic modelling component, we utilize Python NLP libraries: Gensim [12] for all the steps including topic modelling and spaCy [13] for lemmatization. One step that is also customized to the maintenance application is the removal of stop phrases. We use a collection of English stop words pre-defined by Gensim and corpus-specific common phrases such as questionnaire forms repeated across the majority of cases, since question formulations do not characterize individual cases.

One characteristic of LDA model is that it provides different topic distributions depending on a random seed used in its initialization. Therefore, every LDA model with the same set of parameters, except for the random seed, should be computed several times that will be referred to as *runs* further in the text. Afterwards, all the metrics should be averaged over several runs to get consistent results and minimize the influence of the algorithm's stochastic behavior. Another control parameter is the number of topics

which spans from 2 to 20 in our experiments. All the metrics presented in the paper are evaluated at top $K$ retrieved parts, $K = 5, 10$. The algorithm is set up to learn symmetric $\alpha$, a document-topic prior, from data as well as $\eta$, a topic-word prior. The number of iterations is fixed at 100.

In addition, we set an empirical parameter for the ratio of English words appearing in the case description $R_{En} = 30\%$. A topic will be derived by LDA trained on the entirely English corpus in case the description contains at least $R_{En}$ English words, otherwise the maintenance case will be marked as "topic undefined".

## 5. Results and Discussion

In this section, we compare the results of the initial ER architecture evaluation to the results of the modified architecture with the topic modelling component as well as to the best possible results for the dataset of maintenance cases. We group queries by levels of generalization, which stands for the number of matched cases and retrieved parts in our setting. Moreover, since a number of topics is a hyper-parameter that is not learned via training, we discuss the estimation of a possible number of topics using NLP coherence metrics and compare it with observations of the retrieval system's performance.

### 5.1. Retrieval Performance at Top K Parts

The performance of maintenance cases and parts retrieval in the initial configuration of the part retrieval system (Baseline) and the configuration with LDA topic modelling component (LDA) is evaluated using the above described metrics at different $K$. These results are also compared to the best possible results on the test dataset computed at $K = \infty$. We report a 95%-level confidence interval of the mean values of 5 runs with different random seeds for LDA initialization in Figure 3. In addition, we show the ratio of test queries for which the metrics improved with the topic modelling component in comparison to the baseline implementation in Figure 4.

Comparing baseline results at different top $K$ retrieved parts, it can be seen that the values of *success*, *completeness*, *recall* and *min_top_k* increase with higher $K$ and achieve the possible maximum at $K = \infty$. *Min_top_k@$\infty$* is not the target value for this metric, since it is higher than the values of *min_top_k@K* for any $K \neq \infty$ while the goal is to

minimize it. Since we target at the lowest *min_top_k@K* possible, this metric is improved when the average value decreases.

Overall improvement is observed for the experimental configuration with the topic modelling component. For metrics evaluated at $K = 10$, the improvement reached 54.5%, 52.6% and 51.8% of maximum possible improvement for *completeness*, *recall* and *success*. It indicates that the introduced component effectively captures similar cases and therefore parts, too. The performance improvement influenced by topic modelling is more prominent at smaller values of $K$ as can be seen from the difference between the average baseline values of *completeness*, *recall* and *success* and those of LDA in Figure 3.

There is an increase in the ratio of improved queries for *completeness*, *recall* and *success* calculated at smaller $K$ as depiced in Figure 4. For example, from less than 4% of queries for *recall@10* to around 5.45% for *recall@5*. Turning now to the ratio of queries with improved *min_top_k@K*, it is higher for larger $K$ since the set of top ranked parts increases with greater $K$ likewise the probability of finding all of the necessary parts among top $K$ parts. Yet, it is the metric with the most prominent progress according to the ratio of queries that were improved using topic modelling: 10.49% to 11.20% for the LDA configuration.

While for some queries the metrics were improved by the introduction of LDA component, 0.007% to 0.5% of queries experienced deterioration of the *completeness*, *recall* and *success* at different $K$ and 0.8% to 3.2% of queries for *min_top_k@K*. This happens, for example, when a number of documents with the right parts suggestion do not appear in the same group. A possible solution (as well as a future work direction) is to integrate domain knowledge into the system and pre-define the number of topics and their characteristic terms to always appear in the same topic.

### 5.1.1. Performance Evaluation for Queries Grouped Based on the Number of Retrieved Cases and Parts

The queries are grouped by the number of parts used in the query case and retrieved cases as well as by the number of retrieved service cases as demonstrated in Figure A in Appendix. Similarly to Figure 3, the results are reported with the mention of 95%-level confidence interval on average for the runs. We distinguish the queries made for service cases that did not require any part replacement and mark them
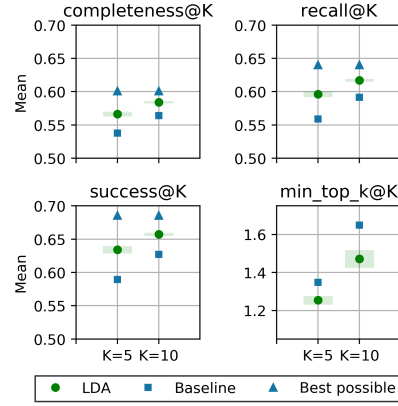


**Figure 3:** Comparison of different metrics computed for LDA and baseline results in a part retrieval task. Confidence interval of 95% is shown as a box around LDA values.
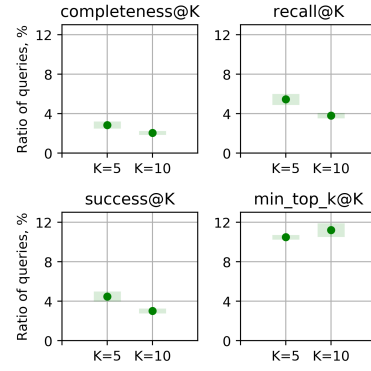


**Figure 4:** Ratio of queries for which the performance metrics improved by the topic modelling component. Confidence interval of 95% is shown as a box around LDA values.

as $|P(c)| = 0$. The groups of queries that benefited the most from the topic modelling component integration are the following:

1. queries with number of retrieved cases $|C(q)| > 100$,

2. queries associated with cases that required $1 \le |P(c)| \le 10$ parts,

3. queries with retrieved and ranked parts $10 < |P_R(q)| \le 100$.

Therefore, the topic modelling has a positive effect on the queries that result in extensive lists of cases and, thus, parts appearing in those cases. Comparing this result to the distribution of queries in our experimental setting (Figure 2), the positive effect concerns the largest groups of queries.
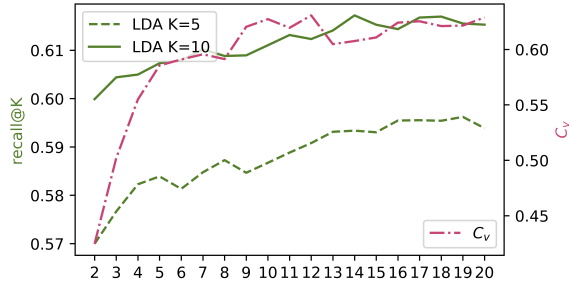
**Figure 5:** Coherence metric $C_v$ and IR metric *recall@K*, $K$ = 5, 10 computed for 2–20 topics derived by LDA.

## 5.2. Number of Topics

LDA requires the number of topics to be passed as an input parameter. In some applications, this value is available as expert knowledge or is motivated by the dataset [14]. Alternatively, a set of coherence metrics could be used to indicate the semantic correspondences within and throughout the derived topics and to evaluate their quality [15]. When a target number of topics is unknown, it could be suggested by the elbow method applied to coherence measures. In our case, the coherence score $C_v$ estimated over 5 LDA instantiations with 2—20 topics resulted in an elbow point between 5 and 9 topics as shown in Figure 5. However, the best results of IR evaluation metrics were obtained in the majority of experiments with LDA at $K$ = 5 for 19 topics and at $K$ = 10 for 14 topics as also demonstrated for *recall@K* in Figure 5. In general, the models perform well with 13 or more topics in our experiment. The impact of the number of topics in terms of chosen evaluation metrics is observed on a smaller scale for 13 or more topics than for the number of topics from 2 to 12.

## 6. Related Work

Areas related to our research span across entity retrieval and knowledge management in industrial applications that correspond to the scope of our work while the use of topic modelling in IR is related to the methodology used in this paper.

### 6.1. Entity Retrieval Overview

Entity retrieval (ER) is defined in [16] as *"the task of answering queries with a ranked list of entities."* The area of entity retrieval is closely connected to IR and

NLP as well as database search and Semantic Web. Both IR and ER are usually enabled with a search engine, a user interface and an available knowledge base. However, while IR aims at document retrieval, the target of ER is to provide a list of ranked entities, such as people, places or specific concepts and things. An entity is characterized with a unique ID, a name and possibly a set of attributes. Data that describes the entity could be stored in natural text or in a more structured form. NLP techniques are used for representation of unstructured texts in a knowledge base, query processing and expansion and query-document modelling. They also facilitate context capturing, named entity recognition, topic-oriented filtering in IR and ER [17, 16, 18]. Considering the classification in [16], our work could be categorized as a study on improvement of an ad-hoc entity retrieval system that uses semantically enriched term representation and preserves topical relations among search results. Examples of industrial entity retrieval often include the knowledge representation in the form of ontology as shown in [3, 4, 5].

### 6.2. Knowledge Management for Industrial Applications

Industries have been adopting process planning and knowledge-based systems for machine manufacturing and maintenance over the recent years [1, 2, 19]. In the literature review on spare part demand forecasting [20], it has been found that a large part of research work has been dedicated to the analysis of historical demand using installed base information and reports.

The work on technical support that utilizes a historical case base is particularly relevant to our research [21, 22, 23]. The goal of the paper [21] is to aid telecom technical support teams with a fast and accurate search over the solutions base for previously registered cases and solutions from other technical texts. A method of populating an existing ontology has been proposed using text segmentation and scoring to serve the use case of Telecom Hardware remote user assistance. The authors in [24] propose a two-step method for spare part demand forecasting that predicts the number of repairs and the number of parts needed for a repair. Our work combines processing of a historical case base, but is not focused on spare part demand forecasting for general planning. It rather considers individual maintenance cases and addresses a lower level of granularity.

***Processing of Technical Documents*** Studies apply NLP as a tool for extracting knowledge from natural texts in industrial log mining [25, 22, 26],

mining technical documentation [27], classification of system failures and preventive maintenance [28, 23].

The study [22] applies an NLP approach to maintenance data concerning a part of the Swedish railway system and identifies frequent failure cases on the railways. Text mining and NLP techniques are applied in [23] to analyze and classify the construction site accidents using the data from Occupational Safety and Health Administration. In this setting, an ensemble method was used to obtain Tfidf matrix and a sequential quadratic parsing method to assign weights to 5 classifiers.

The work [29] focuses on building Machine Learning (ML) models to estimate future duration of maintenance activities by identifying problem, solution and items features via text mining for pre-processing followed by neural networks and decision trees for prediction. NLP is used to mine electronic documents composed of free-form text to extract terms of interest, the hierarchy of their contexts and form a set of normalized terms including multi-word terms for further data analysis in [30].

Therefore, problems addressed in maintenance services application domain are diverse in nature. However, to the best of our knowledge the current paper is the first attempt to use entity retrieval techniques for spare part management.

## 6.3. Use of NLP and Topic Modelling in IR Systems

The effectiveness of IR systems could be improved by topic modelling that mines term associations in a collection of documents. Topic modelling could be integrated to IR tasks to smooth the document model with a document term prior estimated using term distributions over topics [31]. The work [32] explores the possibilities of modelling term associations as a way of related terms integration into document models and proposes a model of probabilistic term association using the joint probability of terms. A combination of term indexing and topic modelling approaches is proposed in [33]. In the proposed model, every query term in a document is weighted using the LDA algorithm and IR indexing methods. The best experimental results were obtained with LDA-BM25 version. However, in this paper, the similarity is computed using a vector space model and the retrieval results are combined using topic relations mined from a historical case base. Therefore, topic modelling is used as a clustering or grouping method on top of an ER system.

In a number of research works, a combination of topic modelling and IR is applied to small texts [34]. For instance, the paper [35] describes a method that first pools similar tweets using an IR approach, merges relevant short texts in a larger document and trains LDA model on concatenated documents thus obtaining richer topics. By contrast, our method addresses a domain-specific collection of short texts written in so-called telegraph style with spelling mistakes and domain-related abbreviations.

***Search Results Clustering*** To date, several studies have investigated document and language models based on topics and clusters. The work [36] explored a cluster-based retrieval of documents, a mechanism that returns a relevant cluster of documents, and proposed two language models for ranking the clusters of documents and smoothing the documents using clusters. By contrast, some works cluster search results using traditional ML, graph-based and rank-based clustering techniques [6, 37]. For instance, Lingo algorithm [38] focuses on learning phrases to represent clusters in a human-readable way and then it discovers topics using Tfidf weighting, performs term-document matrix reduction with SVD and matches the extracted phrases with topics. In comparison to these approaches, our work aims at retrieving entities rather than documents and the user can explore all the retrieved parts within all the clusters instead of only one cluster.

# 7. Conclusion

In this work, we explored a way of improving a spare part retrieval system for remote diagnostics and maintenance of medical equipment by applying topic modelling to search results. The topic modelling component was used to cluster the results of a baseline retrieval system and improve the relevance of the search results. We aimed to support the decision-making process of maintenance service teams that searched in a historical collection of troubleshooting reports and retrieved parts needed for a new similar issue.

The experimental dataset was constructed from query-result pairs pointing at the historical case base and parts used in the cases. We adjusted several IR metrics to evaluate the results of spare part retrieval in the baseline architecture and the topic modelling component modification. The major enhancement was observed for the metric that estimated the minimum top ranked parts that were sufficient for the full treatment of a service case associated with a

performed query.

A natural progression of this work is to apply online topic learning and automatically recommend the topic that performs best for a given query. An input from domain experts would help fix the number of topics and characteristic terms that should appear under one topic. Furthermore, additional domain knowledge could be combined with the entity retrieval system under consideration to suggest actions beyond part replacement, such as troubleshooting tests for remote and on-site diagnostics.

## Acknowledgments

## References

[1] G.-F. Liang, J.-T. Lin, S.-L. Hwang, E. M.-y. Wang, P. Patterson, Preventing human errors in aviation maintenance using an on-line maintenance assistance platform, International Journal of Industrial Ergonomics 40 (2010) 356–367. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169814110000028. doi:10.1016/j.ergon.2010.01.001.

[2] E. Ruschel, E. A. P. Santos, E. d. F. R. Loures, Industrial maintenance decision-making: A systematic literature review, Journal of Manufacturing Systems 45 (2017) 180–194. URL: https://doi.org/10.1016/j.jmsy.2017.09.003. doi:10.1016/j.jmsy.2017.09.003.

[3] Z. Li, K. Ramani, Ontology-based design information extraction and retrieval, Artificial Intelligence for Engineering Design, Analysis fand Manufacturing 21 (2007) 137–154. URL: https://www.cambridge.org/core/product/identifier/S0890060407070199/type/journal{_}article. doi:10.1017/S0890060407070199.

[4] K. Ponnalagu, Ontology-driven root-cause analytics for user-reported symptoms in managed IT systems, IBM Journal of Research and Development 61 (2017) 53–61. doi:10.1147/JRD.2016.2629319.

[5] M. Sharp, T. Sexton, M. P. Brundage, Toward Semi-autonomous Information Extraction for Unstructured Maintenance Data in Root Cause Analysis, in: IFIP Advances in Information and Communication Technology, volume 513,

2017, pp. 425–432. URL: http://link.springer.com/10.1007/978-3-319-66923-6{_}50. doi:10.1007/978-3-319-66923-6_50.

[6] H. Toda, R. Kataoka, M. Oku, Search Result Clustering Using Informatively Named Entities, International Journal of Human-Computer Interaction 23 (2007) 3–23. URL: http://www.tandfonline.com/doi/abs/10.1080/10447310701360995. doi:10.1080/10447310701360995.

[7] S. E. Robertson, S. Walker, K. S. Jones, M. M. Hancock-Beaulieu, Okapi at TREC-3, Proceedings of the Third Text REtrieval Conference (1994).

[8] C. D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, 2008. URL: http://ebooks.cambridge.org/ref/id/CBO9780511809071. doi:10.1017/CBO9780511809071.

[9] C. D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, The MIT Press;, 1999. URL: https://nlp.stanford.edu/fsnlp/.

[10] D. M. Blei, A. Y. Ng, M. T. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[11] Elasticsearch B.V., Elasticsearch, ???? URL: https://www.elastic.co/.

[12] R. Řehůřek, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50.

[13] M. Honnibal, I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017. To appear.

[14] R. J. Gallagher, K. Reing, D. Kale, G. Ver Steeg, Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge, Transactions of the Association for Computational Linguistics 5 (2017) 529–542. URL: https://www.mitpressjournals.org/doi/abs/10.1162/tacl{_}a{_}00078. doi:10.1162/tacl_a_00078.

[15] M. Röder, A. Both, A. Hinneburg, Exploring the Space of Topic Coherence Measures, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15, ACM Press, New York, New York, USA, 2015, pp. 399–408. URL: http://dl.acm.org/citation.cfm?doid=2684822.2685324. doi:10.1145/2684822.2685324.

[16] K. Balog, Entity-Oriented Search, volume 39 of *The Information Retrieval Series*, Springer International Publishing, Stavanger, Norway, 2018. URL: https://eos-book.orghttp://link.springer.com/10.1007/978-3-319-93935-3. doi:10.1007/978-3-319-93935-3.

[17] S. Büttcher, C. L. A. Clarke, G. V. Cormack, Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

[18] Z. A. Merrouni, B. Frikh, B. Ouhbi, Toward Contextual Information Retrieval: A Review And Trends, Procedia Computer Science 148 (2019) 191–200. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050919300365. doi:10.1016/j.procs.2019.01.036.

[19] S. P. Leo Kumar, Knowledge-based expert system in manufacturing planning: state-of-the-art review, International Journal of Production Research 57 (2019) 4766–4790. doi:10.1080/00207543.2018.1424372.

[20] S. Van der Auweraer, R. N. Boute, A. A. Syntetos, Forecasting spare part demand with installed base information: A review, International Journal of Forecasting (2019). doi:10.1016/j.ijforecast.2018.09.002.

[21] A. Kouznetsov, J. B. Laurila, C. J. Baker, B. Shoebottom, Algorithm for Population of Object Property Assertions Derived from Telecom Contact Centre Product Support Documentation, in: 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, IEEE, 2011, pp. 41–46. URL: http://ieeexplore.ieee.org/document/5763435/. doi:10.1109/WAINA.2011.135.

[22] C. Stenström, M. Aljumaili, A. Parida, Natural language processing of maintenance records data, International Journal of COMADEM 18 (2015) 33–37.

[23] F. Zhang, H. Fleyeh, X. Wang, M. Lu, Construction site accident analysis using text mining and natural language processing techniques, Automation in Construction 99 (2019) 238–248. URL: https://doi.org/10.1016/j.autcon.2018.12.016. doi:10.1016/j.autcon.2018.12.016.

[24] W. Romeijnders, R. Teunter, W. Van Jaarsveld, A two-step method for forecasting spare parts demand using information on component repairs, European Journal of Operational Research (2012). doi:10.1016/j.ejor.2012.01.019.

[25] R. Sipos, D. Fradkin, F. Moerchen, Z. Wang, Log-based predictive maintenance, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14, ACM Press, New York, New York, USA, 2014, pp. 1867–1876. URL: http://dl.acm.org/citation.cfm?doid=2623330.2623340. doi:10.1145/2623330.2623340.

[26] S. Agarwal, V. Aggarwal, A. R. Akula, G. B. Dasgupta, G. Sridhara, Automatic problem extraction and analysis from unstructured text in IT tickets, IBM Journal of Research and Development 61 (2017) 41–52. doi:10.1147/JRD.2016.2629318.

[27] K. Richardson, J. Kuhn, Learning semantic correspondences in technical documentation, ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) 1 (2017) 1612–1622. doi:10.18653/v1/P17-1148.

[28] K. Arif-Uz-Zaman, M. E. Cholette, L. Ma, A. Karim, Extracting failure time data from industrial maintenance records using text mining, Advanced Engineering Informatics 33 (2017) 388–396. URL: http://dx.doi.org/10.1016/j.aei.2016.11.004. doi:10.1016/j.aei.2016.11.004.

[29] M. Navinchandran, M. E. Sharp, M. P. Brundage, T. B. Sexton, Studies to predict maintenance time duration and important factors from maintenance workorder data, in: Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM, 2019. doi:10.36001/phmconf.2019.v11i1.792.

[30] A. Kao, N. B. Niraula, D. I. Whyatt, Text mining a dataset of electronic documents to discover terms of interest, 2020.

[31] L. Azzopardi, M. Girolami, C. van Rijsbergen, Topic based language models for ad hoc information retrieval, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), volume 4, IEEE, 2004, pp. 3281–3286. URL: http://ieeexplore.ieee.org/document/1381205/. doi:10.1109/IJCNN.2004.1381205.

[32] X. Wei, W. B. Croft, Modeling Term Associations for Ad-Hoc Retrieval Performance Within Language Modeling Framework, in: Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 52–63. URL: http://link.springer.com/10.1007/978-3-540-71496-5{_}8. doi:10.1007/978-3-540-71496-5_8.

[33] F. Jian, J. X. Huang, J. Zhao, T. He, P. Hu, A Simple Enhancement for Ad-hoc Information Retrieval via Topic Modelling, in: Proceedings of

the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16, ACM Press, New York, New York, USA, 2016, pp. 733–736. URL: http://dl.acm.org/citation.cfm?doid=2911451.2914748. doi:10.1145/2911451.2914748.

[34] J. Qiang, Z. Qian, Y. Li, Y. Yuan, X. Wu, Short Text Topic Modeling Techniques, Applications, and Performance: A Survey, IEEE Transactions on Knowledge and Data Engineering 14 (2020) 1–17. URL: https://ieeexplore.ieee.org/document/9086136/. doi:10.1109/TKDE.2020.2992485.

[35] M. Hajjem, C. Latiri, Combining IR and LDA Topic Modeling for Filtering Microblogs, in: Procedia Computer Science, 2017. doi:10.1016/j.procs.2017.08.166.

[36] X. Liu, W. B. Croft, Cluster-based retrieval using language models, in: Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04, ACM Press, New York, New York, USA, 2004, pp. 1–8. URL: http://portal.acm.org/citation.cfm?doid=1008992.1009026. doi:10.1145/1008992.1009026.

[37] K. Sadaf, Web Search Result Clustering- A Review, International Journal of Computer Science & Engineering Survey 3 (2012) 85–92. URL: http://www.airccse.org/journal/ijcses/papers/3412ijcses07.pdf. doi:10.5121/ijcses.2012.3407.

[38] S. Osiński, J. Stefanowski, D. Weiss, Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition, in: Intelligent Information Processing and Web Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 359–368. URL: http://link.springer.com/10.1007/978-3-540-39985-8{_}37. doi:10.1007/978-3-540-39985-8_37.

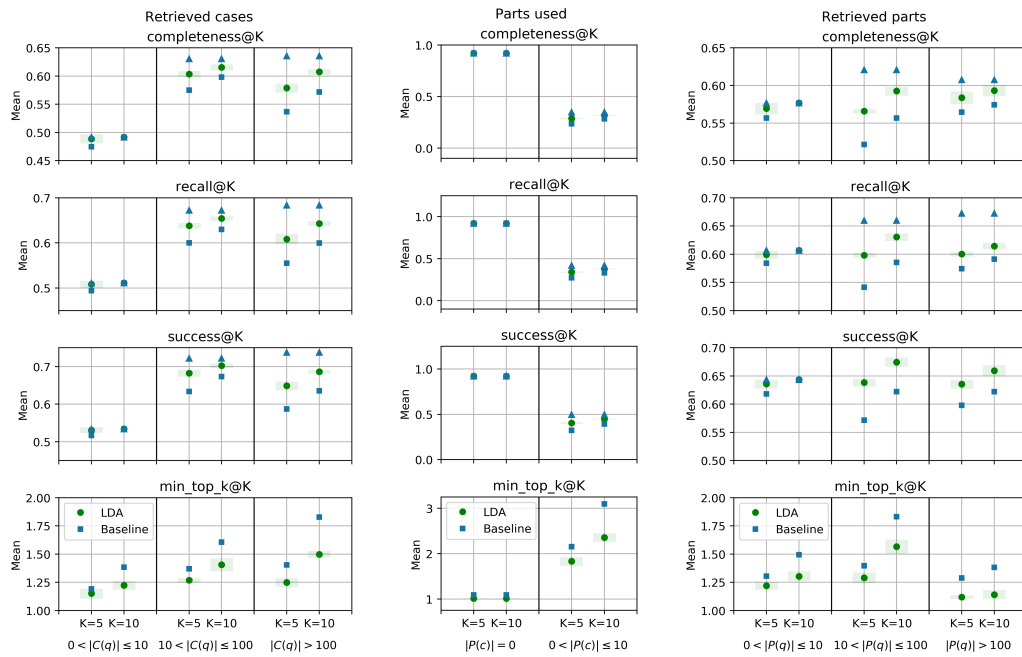# A. Topic Modelling Component Performance Evaluation for Grouped Queries

**Figure 1:** Comparison of different metrics computed for LDA and baseline results in a part retrieval task. Queries are divided into groups using the number of retrieved cases, as well as used and retrieved parts. Confidence interval of 95% is shown as a box around LDA values.