Santeri Asikainen

Experimenting the use of a Large Language Model to improve company name entity resolution

Experimenting the use of a Large Language Model to improve company name entity resolution

Santeri Asikainen
Final projects
Spring 2024
Modern Software and Computing Solutions
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Master’s Degree, Modern Software and Computing Solutions

Author: Santeri Asikainen
Title of the thesis: Experimenting the use of Large Language Model to improve company name entity resolution
Thesis examiner(s): Ilpo Virtanen
Term and year of thesis completion: Spring 2024 Pages: 50

This thesis was commissioned by a software company operating in procurement analytics business which sells software-as-a-service products to customers. One of the key services which is built inside many of the software products is the deduplication of company names, which exist inside and across client databases. Case company wanted to have better understanding of the current system performance, what text-patterns lead to manual corrections and what solutions could help in improving the system performance.

The study was conducted by first getting understanding of the current system and revealing potential issues in the process. Exploratory data analysis was done for two datasets. First dataset consisted of manually corrected entries on the most granular supplier level. Second dataset consisted of supplier groups in the current system which the system had failed to group together. Patterns of both datasets were grouped together and estimated how large of an impact could be achieved by handling or removing the patterns which have caused the mistakes. The analysis revealed that the most impactful pattern is the company name itself, while the impact of other patterns is small. One of the main reasons for the identified mistakes was the deterministic rule-based approach to link two text strings together as it led easily to false non-matching decision. As the Large Language Models have shown rapid improvement in various tasks in recent years, the study continued by experimenting LLM capabilities in recognizing and extracting company name from text strings.

The experiment was done by using OpenAI gpt-3.5-turbo model. The model was given three sample datasets, and the model output performance was compared against skilled human doing the same thing. Quantitative results were calculated as the % of correct responses. The model performance was also analysed using qualitative methods, where the interest was specifically in what kind of answers the model is giving when it fails in the task and is the model ‘hallucinating’ responses.

The model performance was 93.67 % for the dataset which have not yet been processed by case company’s entity resolution process and 70-80 % for the datasets which have already been processed. Case company could improve the current system performance by introducing LLM to the entity resolution process. LLM identification of the company names would help in finding duplicates within existing groups and also in proposing key words for matching for the new data entering the system.

.

Keywords: entity resolution, large language model, prompt engineering, deduplication

CONTENTS

1 INTRODUCTION

1.1 Entity resolution problem

Businesses, governments, and scientific organizations increasingly rely on massive amounts of data collected from both internal and external sources (e.g. CRM, ERP, Web). Combining and linking together this information is highly beneficial resulting in richer and more reliable conclusions and applications in various research disciplines and business domains. However, in most times the same real-life entities cannot be linked together using unique identifiers since they don't exist. Hence, the question of *how to identify and link records that correspond to the same entity within one or across several datasets* has been and still is an unsolved research problem. To tackle the issue, practitioners and researchers have turned to methods from statistics, computer science and machine learning calling the problem as entity resolution, record linkage or deduplication. (Binette & Steorts, 2022; Christen, 2019; Christophides et al., 2020).

As listed by Binette & Steorts entity resolution problem is difficult because of three things that needs to be balanced. 1. Efficient methods which scale to large datasets, 2. Accurate and generalizable methods which make most out of the available information and 3. Appropriately quantifying and propagating the uncertainty which is coming from all stages of the entity resolution process. For practical application, there is typically a trade-off between accuracy and efficiency, or between accuracy and modelling complexity. Despite many attempts to create one method to tackle all the difficulties, the suitability of given entity resolution approach remains highly application-specific.(Binette & Steorts, 2022)

Linking company names together from multiple sources is one application area where there are clear benefits from successful entity resolution. Large companies typically have multi-ERP landscapes with multiple vendor master records resulting in that one supplier company entity may be found in multiple ERPs or created many times within one ERP. This happens particularly with the vendors who are large suppliers for the customer. The records are not duplicates as such but refer to different legal entities or country organizations of a larger group. When reporting global spending, negotiating contracts with suppliers or monitoring contract compliance, usually one would want to see all the suppliers belonging to one group in aggregated manner to make better decisions or draw conclusions in terms of overall business done with the supplier. Without entity resolution, this is cumbersome because in most cases there are no unique identifiers across databases or even inside databases to perform aggregation over the different legal entities or country organization as they are usually maintained in free text fields and hence, written differently by the users creating and maintaining the records.

1.2 Large language models and entity resolution

In recent years, large language models have shown remarkable capabilities in text understanding and generation, allowing for generalization across various domains and tasks. One application area has been information extraction from unstructured text to structured knowledge, which is crucial domain in natural language processing as it is often foundational requirement for many downstream tasks such as question answering or entity resolution. (Xu et al., 2023)

One of the information extraction tasks that large language model can be used in is Named Entity Recognition, which refers to recognizing and extracting entity from a given text. The major benefit of LLMs compared to pre-trained language models is the unnecessity of large amounts of task specific training data. To get optimal results from the LLM, prompt engineering has become a critical factor, as it dictates to large extent what kind responses can be expected from the general purpose LLM. (Brinkmann et al., 2024; Xu et al., 2023)

1.3 Logic and purpose of the thesis

The thesis subject has originated from case company's need to understand how well the supplier service entity resolution system is working and how to better identify the problematic areas and improve performance. Hence, the first research question is:

- What patterns in the data or process lead to incorrect automatic results in the case company's entity resolution system?

As the LLMs are not incorporated in the current process of case company's system, second research question aims to answer if this new tool should be considered:

- Could the use of LLM improve the performance of the current system and in what ways?

1.4 Structure of the thesis

The first chapter in the thesis focuses on introducing the case company, area of business and how the current process works. The case company's process is compared to a typical entity resolution (ER) workflow found in the literature. The second chapter focuses on exploratory analysis, which aims to draw insights on how the data looks like and identify if there are any conclusions that can be drawn from the data. One of the key elements of the chapter is to identify problematic patterns in the data and use this information in the upcoming LLM experiment phase. Third chapter introduces briefly the LLMs in information extraction and the concepts of zero-shot and few-shot prompt engineering. The fourth and fifth chapters are focusing on the experiment setup and results. Finally, conclusions and discussion wrap it all together.

2 CASE COMPANY AND SUPPLIER SERVICE

2.1 Company overview and supplier service

Case company operates in software analytics business and has a client base of large multinational corporations. Company is providing software-as-a-service analytics platform using invoice and purchase data from clients' source systems as the base data for analytics solutions and services. Source data includes vendor records, which are consolidated inside a tenant as well as globally to the central supplier repository. Consolidation here means same as entity resolution – identifying real world entities which should be grouped together. In below example (Figure 1), the customer 1,2,3 illustrates the starting point, i.e. the source data. In typical cases, inside each customer source data database, the entity resolution has not happened yet and therefore the source data fed into the supplier service has duplicate entries inside and between customers from real world company group point of view. This is apparent specifically in the cases of large known global suppliers providing products and services used in fundamental business operations, such as DHL in logistics.

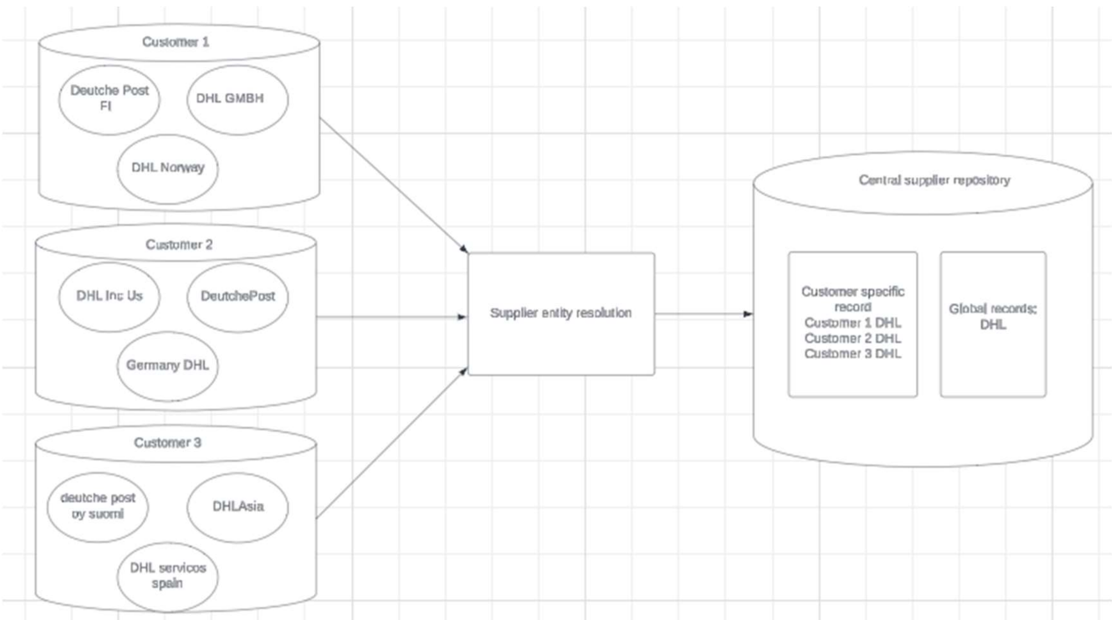


Figure 1. Illustration of supplier service in case company.

2.2 Comparing case company entity resolution workflow with typical ER workflow

Typical entity resolution workflow comprises of 5 elements or process steps. 1. Once the data has been extracted from source systems it needs to be cleaned/standardized, 2. Indexing the data for later processing 3, comparing the potential matches 4. Classifying the data and finally 5 evaluation. (Hand & Christen, 2018)

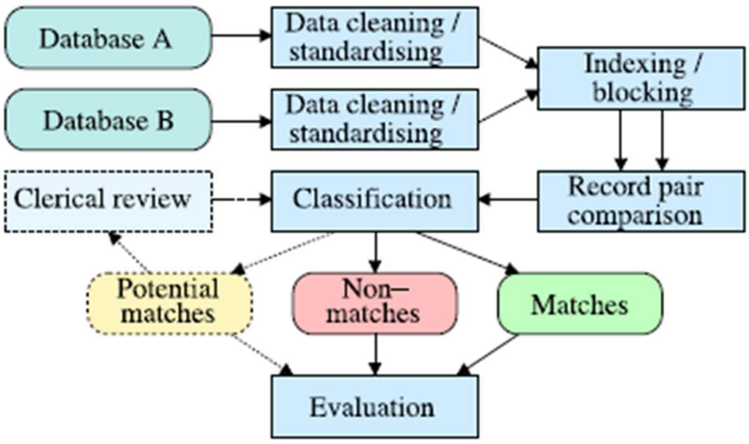
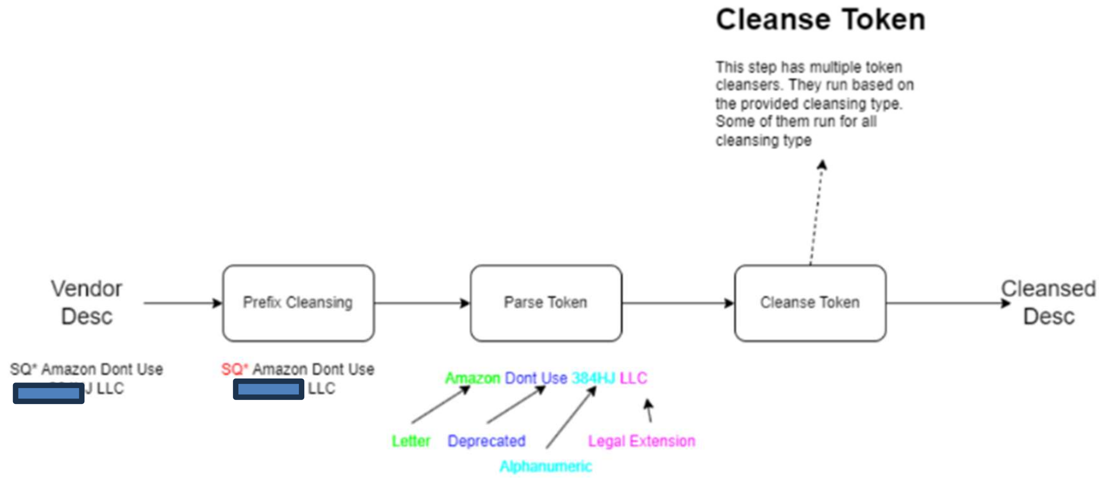


Figure 2. Typical ER workflow (Hand & Christen, 2018)

2.2.1 Data cleaning/standardising

The first step in the entity resolution process is data cleaning/standardising. This step ensures that the attributes (data fields) which are later used in comparison are consistent and in same formats. This processing includes for example converting all letters to lower-case characters, expanding abbreviations and replacing known misspellings. (Hand & Christen, 2018).

In the case company ER process, many steps are used to clean and standardize the data input. Below picture illustrates how the cleaning process works.



The Vendor Desc is the raw source data entry from the client system. The cleaning algorithms are checking and removing defined prefixes which are considered as not relevant for the matching. In addition, certain tokens often found in the source data (such as Don't Use) and legal extension (e.g. LLC) are removed. The final Cleansed Desc is then the normalized name which is later used in the Record pair comparison and matching phase. Different matching algorithms may have different cleaning logic.

2.2.2 Indexing / blocking

Indexing or blocking is a step that is aiming to group together records that are potential matches to improve scalability and computing performance. Instead of comparing all the records together in the database in the matching phase, the blocking first filters the candidates to smaller datasets and the matching comparison is done inside these blocks. Without blocking, the number of comparisons increase quadratically with the amounts of records in the database and may lead to performance bottlenecks. (Hand & Christen, 2018)

Currently there is no blocking phase in the case company's entity resolution workflow, mainly because the matching is happening over exact matches.

2.2.3 Record pair comparison and classification

Record pair comparison is the process step where record pairs are evaluated to distinguish whether they are the same entity or not. Depending on the comparison results, pairs are either classified as match or non-match or something in between. There are various approaches to decide the matching. On high-level these approaches can be categorized to deterministic and probabilistic methods, which treat the entity resolution as classification problem and clustering methods which see the ER as a clustering problem. Deterministic approach is the most commonly used approach in practice, and it is based on rules. Example of deterministic matching in its simplest form is exact matching, where two records are linked if they agree on all common attributes. There is no room for ambiguity or partial matches in the deterministic approach. However, often in real life scenarios the data is not perfect and the best that can be obtained is a probability that the records match, maybe match or do not match. This is the probabilistic approach which gives a probability score as an output to decide on the linking. Typically, a similarity score is calculated between the compared attributes in the records and the scores are stored in a similarity vector for the pair. The scores of the similarity vector are combined to get a single value between 0 and 1 and if a certain threshold is met, the records are linked together. The third set of approaches, where entity resolution is seen as clustering problem, aims to tackle the problem of transitivity which is present in record pair-wise comparison methods. Transitivity is present specifically when linking two or more databases where duplicates exist within and across databases. Transitivity means that if we have 3 records A, B and C, where A and B link together and B and C link together, then automatically A and C should also link together. In pairwise comparison this is not always the case. (Binette & Steorts, 2022)

Case company record matching phase is based on deterministic approach. Depending on the algorithm, the attributes of a new supplier entering to the system are compared with existing attributes in the existing supplier groups. If a match is found, a linking is suggested. Confidence parameter for the matching is calculated, but it is not a probabilistic similarity score between the record pairs. The higher the confidence level, the more manually validated entries exist in the group where the matching attribute is same. In addition, the confidence level depends also on if the match is found inside or outside the same customer. Different algorithms are given different kinds of minimum and maximum ranges for the confidence score. However, even if an existing group is not found, i.e. a new supplier group is created to the system, the confidence is above 0 given that at least one of the algorithms are giving a suggestion to create a new group.

2.2.4 Evaluation and review

Entity resolution evaluation can be illustrated with below confusion matrix. (Hand & Christen, 2018)

		True link status	
		Match	Non-match
Predicted	Match	d (true match)	b (false match)
link status	Non-match	c (false non-match)	a (true non-match)

Figure 3 - Predicted vs True link status matrix (Hand & Christen, 2018)

In binary scenario, record pair will be predicted as Match or Non-match by the system (predicted link status above). The true link status however can have 4 different statuses. 1. True match, which is the correct match predicted. 2. False match, where the true status is non-match but is predicted as match. 3. False non-match, where the true link status is match, but is predicted as non-match and finally 4.true Non-match, where the true link status as well as predicted status is a non-match. When above confusion matrix is defined, F1 score can be calculated to estimate the true performance of the entity resolution system. F1 score is by far the most used evaluation metric to assess the goodness of entity resolution systems. Reason for the use of F1 as an estimator is the large imbalance between categories. In typical scenario, the amount of non-matches exceed the number of matches massively and hence using a simple accuracy as the error rate would provide meaningless results as just classifying everything to non-matches would end up with 99,9 % accuracy in most real life entity resolution cases. (Christophides et al., 2020; Hand & Christen, 2018)

Evaluation of the goodness of the system in case company is happening currently with the use of manually validated entries. For each customer, largest suppliers are reviewed manually, and the automatic grouping is changed if it is considered incorrect. By comparing the total amount of manually reviewed groups with the number of changes that have been done inside the manually reviewed groups, an accuracy metric is obtained. The accuracy does not make distinction between the 4 categories above, meaning that it is not possible to currently know if the reason for change was due to the false non-match or false match. However, as the current matching logic is deterministic and requiring exact matches from the comparison attributes, the working hypothesis is that majority of the corrections are due to the false non-match category.

2.3 Overview of parenting algorithms

In current process, case company has multiple algorithms which are used to find attributes to determine the matching. Table 1 describes on high-level how the algorithms are used.

Table 1. Parenting algorithms.

Alogorithm	How it is used
VAT	Finds if same VAT code is used in other entries in system, if yes, match is proposed
Vendor Enrichment	Uses third-party database to check if same name exists there, if yes, suggests a match based on the existing record
Employee	Finds ERP supplier records which have been flagged as employee records, matches all employee records to same supplier group
Keyword	Users can give manually key words. If the key word is found inside the text string of new record, record is matched into the existing group by keyword algorithm.
Migration	Uses legacy supplier groups (entries which have not been ran through current service) as a suggestion for grouping.
Description	Uses cleansed supplier descriptions to check if same cleansed name is found in existing system, if yes, match is proposed
Payment Services	Looks for predefined prefix in the beginning of supplier text string, if found, suggests parenting based on the predefined prefix.
Undetermined	Used only when no other algorithm is giving suggestion for grouping, helps to identify records which have not been parented to any supplier group.

2.4 Summary of the current process in case company

Below Figure 4. describes how the entity resolution workflow works in the case company for new, unseen data. The deterministic approach has its pros and cons. In a nutshell, the selected approach minimizes the risk of false matches at the expense of false non-matches. The impact of the current approach will be discussed more in chapter 3 – Explorative analysis.

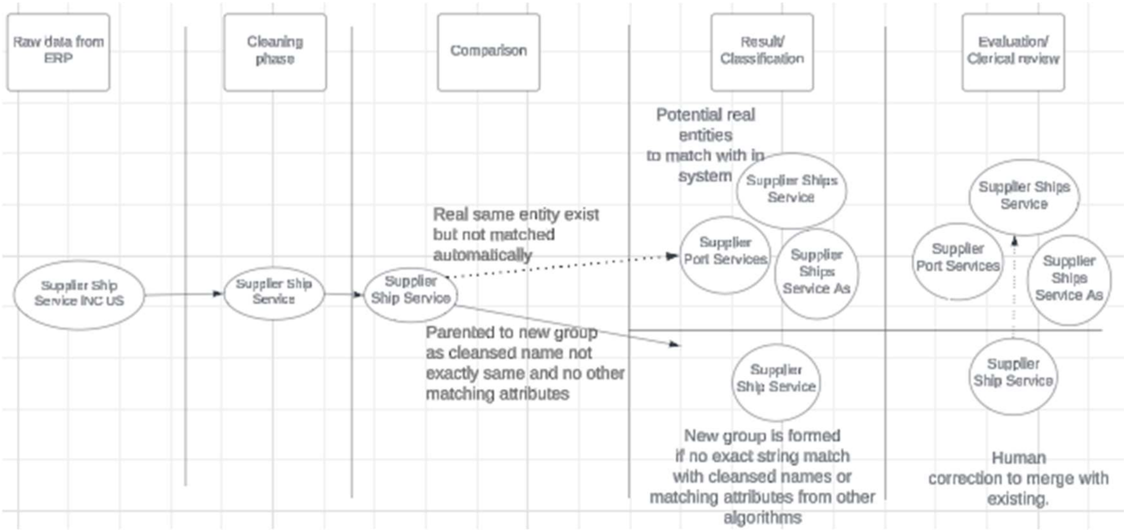


Figure 4. Illustration of the case company's entity resolution process for unseen data.

3 FINDINGS FROM EXPLORATIVE ANALYSIS

Explorative analysis is used to get overall understanding of the data and what is happening. First goal of the analysis was to get idea of the current supplier group statistics. Second objective was to identify if there are any patterns emerging in the data which are hindering the entity resolution performance.

3.1 Statistics of current system

Explorative data-analysis was conducted to identify the pain points in the current system. First, the main statistics (Table 1) of the system were generated to see the overall picture.

Table 2. Supplier group statistics (retrieved from database in February 2024).

Metric	Value
Distinct Supplier Groups	5 591 666
Average Group Size (Global) in number of suppliers	1,9
Average Group Size (Tenant) in number of suppliers	1,5
Distinct Group Sizes (Global)	757
Maximum group size (Global) in number of suppliers	128 074
Frequency for Group Size = 1 Supplier (Global)	74 %
Frequency for Group Size = 2 Suppliers (Global)	16 %
Count of duplicate Supplier Group Names (Global)	604
Tenant (customer) count	63
Shared Supplier Groups between Tenants (Supplier Group in more than one Tenant)	10.5 %
Average % of Supplier Groups shared between 2 Tenants	0,17 – 0,47 %

The difficulty of analysing the results from above was that there are no benchmark figures available. Hence, it is hard to say what is a good level of grouping even without considering the quality of grouping. Still, the hypothesis was that the average group size seemed to be rather small. Also, when sample supplier groups were checked in the system, it seemed that there are multiple supplier groups existing for the same real-life entity which indicates that there is room for improvement in the process. For example, a major supplier was having 600+ supplier groups in the system relating to one real life entity. However, given the deterministic strict matching rules, the result was somewhat expected. Reason for the strict matching rules is the fact that from business and usability point of view, it is easier to merge false non-match to existing group than demerging false match from existing group in the manual review process. In addition, from quality and analytics use case point of view, having suppliers wrongly grouped together generates more questions and unsatisfaction compared to having too many groups for a given real life entity.

Another goal for the general statistics check was to find out which of the parenting algorithms have the highest impact i.e. how often a decision from a certain algorithm is selected compared to others.

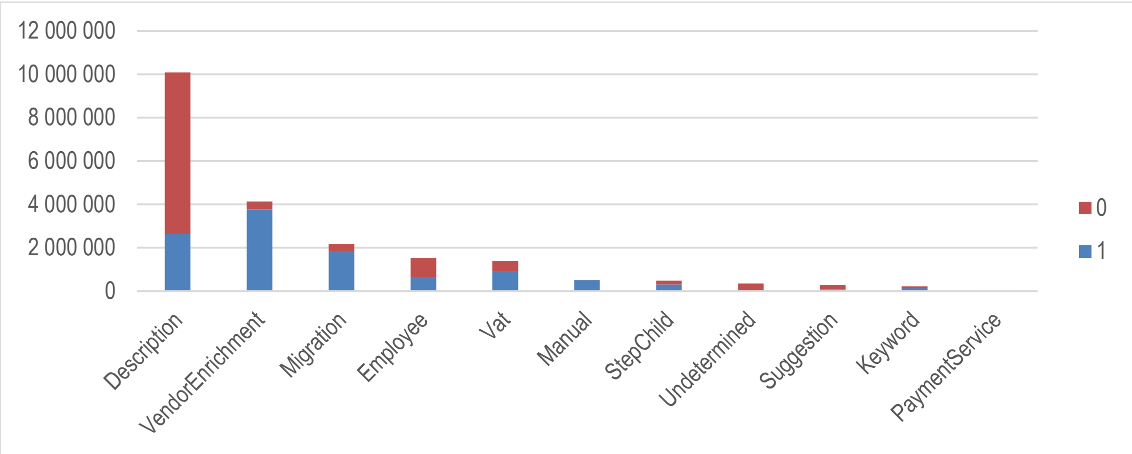


Figure 5. Row distribution of parenting algorithms.

Figure 5. highlights that the Description and Vendor Enrichment parenting algorithms are by far the most used ones. The 1 in the stacked bar indicates that there are purchase transactions attached to the supplier and 0 that there isn't. From business use case point of view, this distinction is important, as it is more impactful to improve the groupings of the suppliers which have transactions involved.

Second objective of the analysis was to dig deeper in the errors to understand better why certain automatic matching decisions lead to manual corrections and if there are some clear patterns among these. This analysis was divided into two parts to obtain good understanding of the potential patterns. First part of the analysis was conducted only for the results of incorrect 'Description' parenting, mainly because it is among the largest algorithms and because Vendor Enrichment algorithm is partly a third-party solution for which case company does not have full control over. Second part of the analysis was done for randomly selected large companies inside Fortune 500 list to understand why the companies are having multiple supplier groups in the system

3.2 Pattern analysis for manually corrected entries

Data was selected randomly for 700+ supplier groups which were manually changed. The dataset consisted of the initial supplier group proposed by the algorithm as well as the current supplier group to which it was changed into in the clerical review process. The differences between the original and current supplier group were reviewed and the errors/differences were categorized. The error patterns were calculated together and when possible, checked how often similar patterns occur in the database among current supplier groups.

The first identified errors were grouped to be originating from the cleaning phase and improvements and alignment here could be beneficial and would not require changing existing matching logic or creating new algorithms. Below table illustrates the found errors.

Table 3. First group of patterns found in manual correction analysis.

Pattern	Relating to	Reason	Examples
Company abbreviations, ‘Company’ or ‘Corporation’ endings in the end of string	Cleaning phase	In cleaning phase, not all company abbreviations are removed or they are handled differently between algorithms	’Co’, ’Sa’, ’Sas’, ’Corporation’, ’Ld’, ’Est’, ’INC’, ’Ab’, ’Kg’, ’Lt’, ’Oy’, Kommanditgesellschaft, Company, ’limi’
Country/location ending or abbreviation	Cleaning phase	In cleaning phase, not all country and location abbreviations are removed or they are handled differently between algorithms	Star Bucks Coffee Japan, Omilo Sweden, Pharma Graphic Canada

Handling of 'Group' and abbreviations of groups	Cleaning/matching phase	Sometimes Group is in the name and sometimes not. Strings don't match if Group is in another name and not in another.	Schunk Group vs Schunk Publicis Group vs Publicis
Handling of special characters and numbers in cleaning	Cleaning phase	Some algorithms are removing hyphons, numbers in the end etc. and others are not	Wendys vs. Wendy's Exxonmobil 45302045 vs Exxonmobil

The second group of patterns are concerned with having similar naming, but not directly something that could be easily removed by defined rules. Improving the grouping in this group would require changes to the matching logic or to define new algorithms.

Table 4. Second group of patterns found in the manual correction analysis.

Pattern	Relating to	Reason	Examples
Clearly same entity but different wordings and words	Matching logic	As matching is happening over deterministic exact string matching, differences are resulting to a new group	Bowmill Group vs. Bowmill Metal Treatments Publicis Group vs. Publicis Communications Johnson Matthey vs. Johnson Matthey Davy Technologies Aramco vs.

	Saudi Aramco Oil Company

The third group is the most difficult one from entity resolution point of view as one cannot interpret from the name itself if the suppliers should belong together or not and the grouping must rely on other attributes than a name text string for automatic grouping.

Table 5. Third group of patterns found in the manual correction analysis.

Pattern	Relating to	Reason	Examples
Group names	-	Companies should be under same entity but this cannot be direcly interpreted through text strings	Travelocity vs. Expedia Group Hampton Inns vs. Hilton Chevron Corporation vs. Miller Industires

3.3 Pattern analysis for sample of Fortune 500 companies

When checking current groupings in the system it became evident that there is substantial number of duplicate groups. Below illustration shows couple of companies and their current groups, the larger the bubble the more suppliers are belonging into that group. The brackets in the picture indicate the count of suppliers and Keywords: No/Yes if keywords are in use or not. The locations of the bubbles in the picture are random.

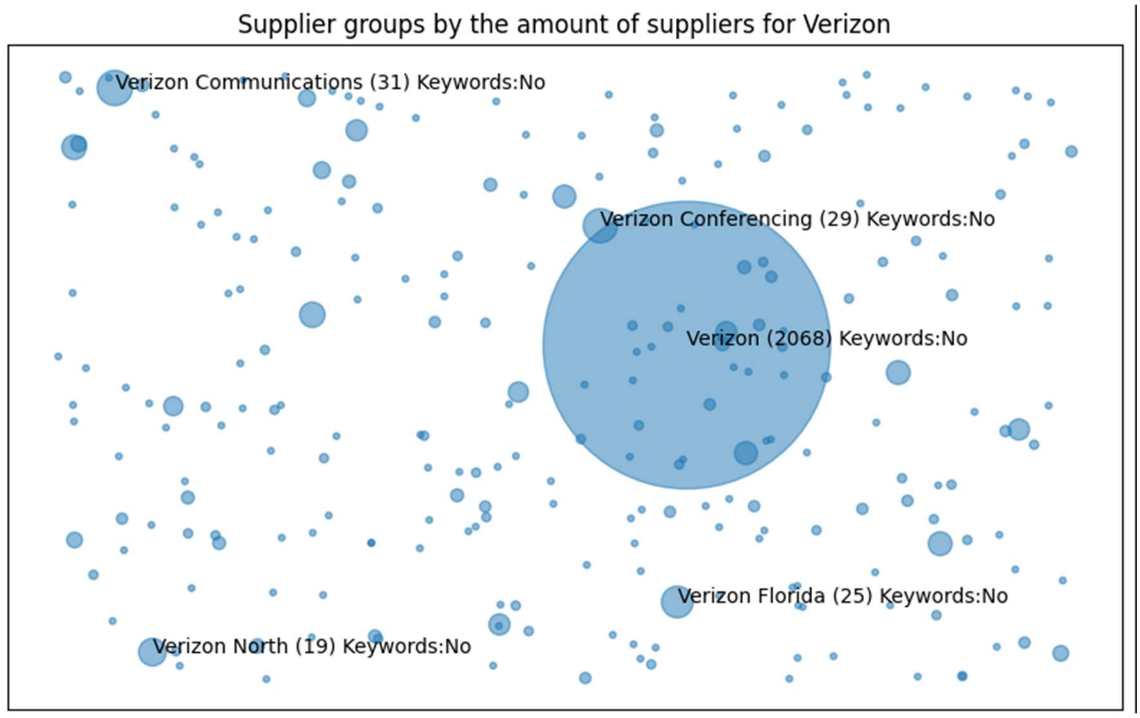


Figure 6. Example supplier groups in system.

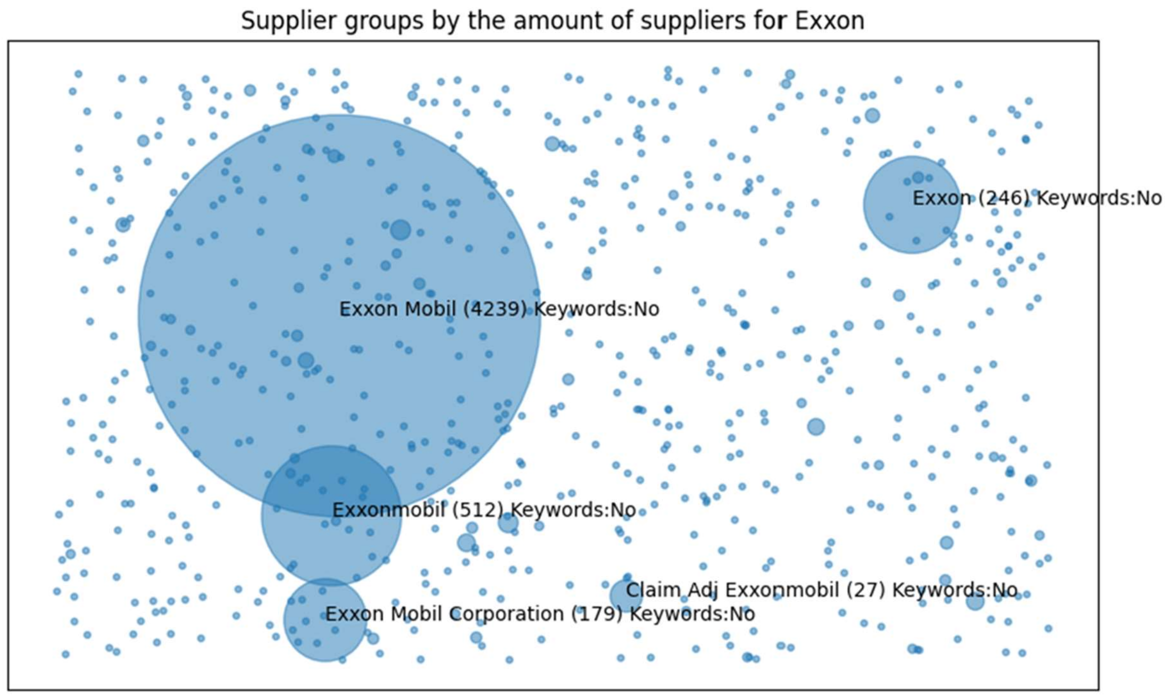


Figure 7. Example supplier groups in system.

Among the selected 12 companies, which in best case scenario would have 12 supplier groups in the system, totaled to over 2000 groups. When analyzed further, one clear differentiator showed up between the companies: Keywords. If keywords were in use, the number of groups decreased substantially.

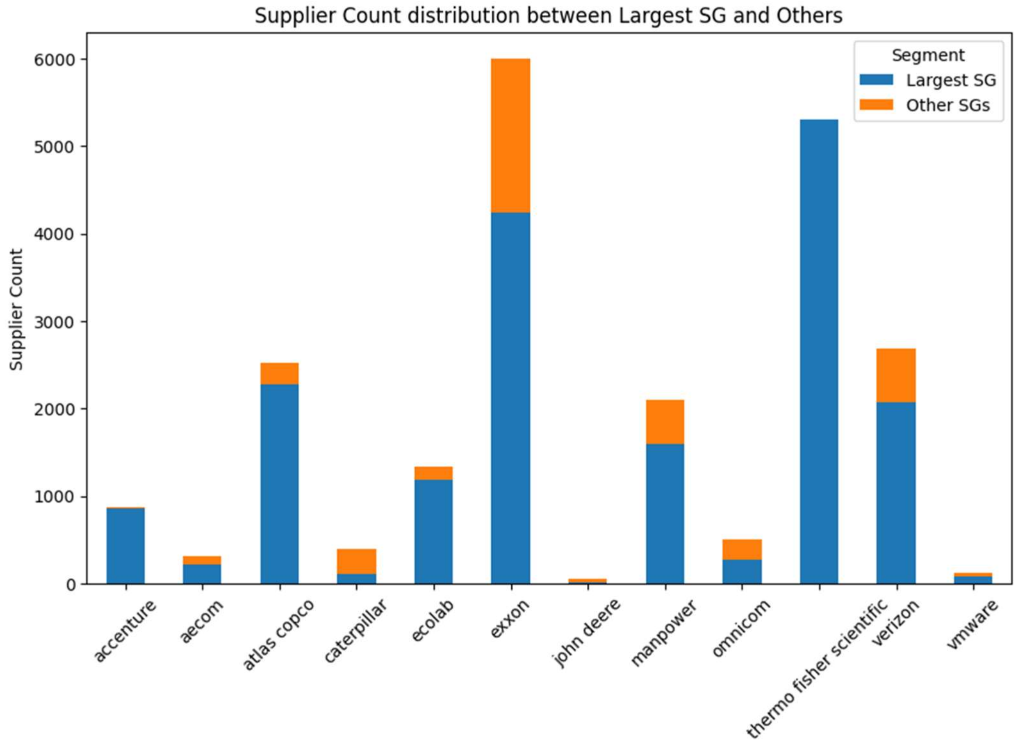


Figure 8. Supplier counts for companies between largest group largest group and others.

The above picture displays the distribution of largest supplier group compared to others within that company by the number of suppliers. Almost all the companies are having one big group in the system and varying number of suppliers in other groups. The best performing companies from deduplication point of view seem to be accenture, thermo fisher scientific. The commonality between these is the use of keywords. Keywords mean words which are searched within the company name text strings by keyword algorithm, and if found, match is proposed.

In addition to how the groups look like in the system, actual patterns from the supplier group text strings were examined. Word cloud was formed from the supplier group string tokens after removing the actual company name to check if there are any clear patterns emerging. There were not any major patterns which could be considered as relevant or where more emphasis should be placed on.

Value	Count	Frequency (%)
c/o	147	3.5%
global	145	3.4%
mi	141	3.3%
services	79	1.9%
	59	1.4%
s	40	0.9%
charge	38	0.9%
o/s	38	0.9%
back	38	0.9%
group	35	0.8%
Other values (1909)	3479	82.1%

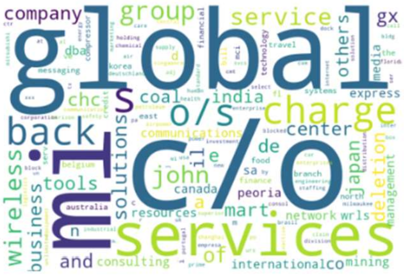


Figure 9. Frequency of remaining tokens and word cloud after company name removed from name string.

After the token frequency, manual categorization of remaining string tokens was performed. Remaining string tokens here means the text string which is left when the company name is removed. After initial look at the data and findings from the first pattern analysis, the remaining strings were grouped into six categories: Location, AllRemainingTokensHaveMeaning, RemainingStringContainsTokensWithoutMeaning, Company/Group abbreviation, OtherLanguage and DataMistake. The counts of the categories are illustrated below:

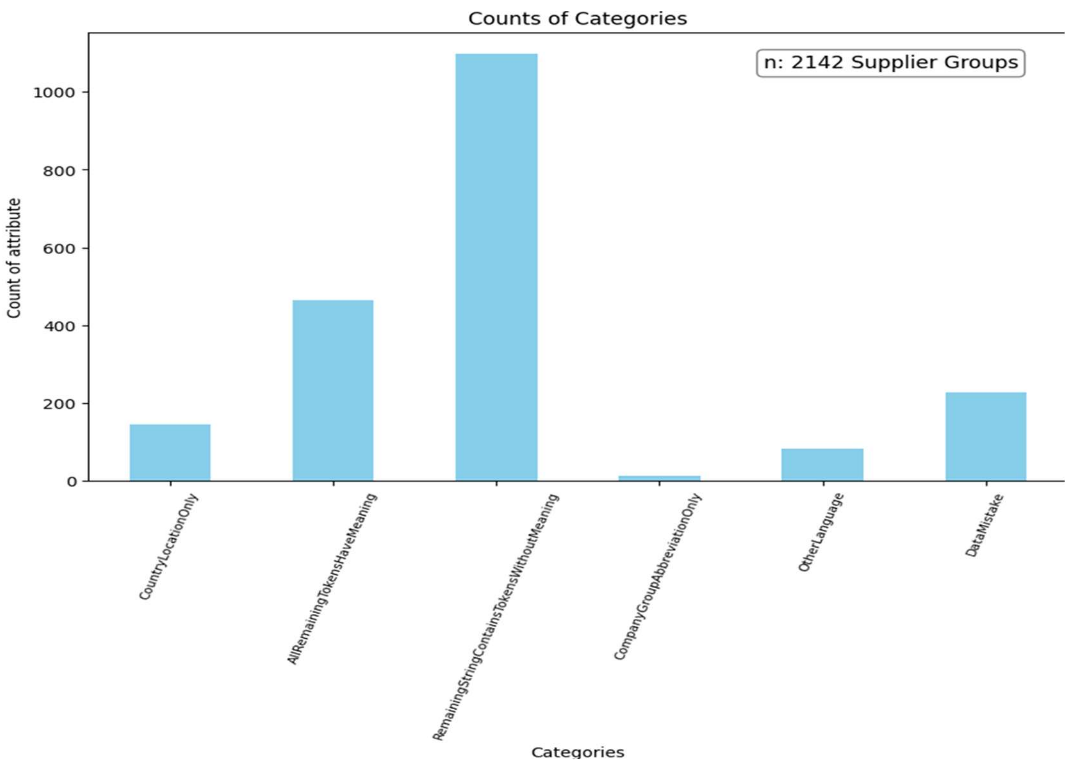


Figure 10. Manual categorization of remaining string tokens after company name removed.

Examples of the values in each category are: Location: 'Verizon **Switzerland**'; All remaining tokens have meaning: 'Manpower **Enterprise Management**'; Remaining string contains tokens without meaning: 'Caterpillar **Logistics Serv**'; Company group abbreviation only: 'Exxon **Co**'; Other language: 'Manpower **Planen U Leisten**'; Data mistake: '**Southwest Airlines**'. The reason for the distinguishment between if a token has a meaning or not was that for potential future fine tuning, this information may be useful. Data mistake in the sample may be due to two reasons, 1. either the supplier's name which was used to pull the supplier groups was categorized to wrong group in system or 2. The supplier was categorized into a correct group but it was not actually inside the twelve companies which were examined due to the method the sample was formulated.

Other findings from the data illustrated below were that 1. The actual company name appeared first in the string in roughly 80 % of the cases. 2. Most of the remaining strings have 1-3 tokens and 3. The remaining strings typically have more than 5 characters.

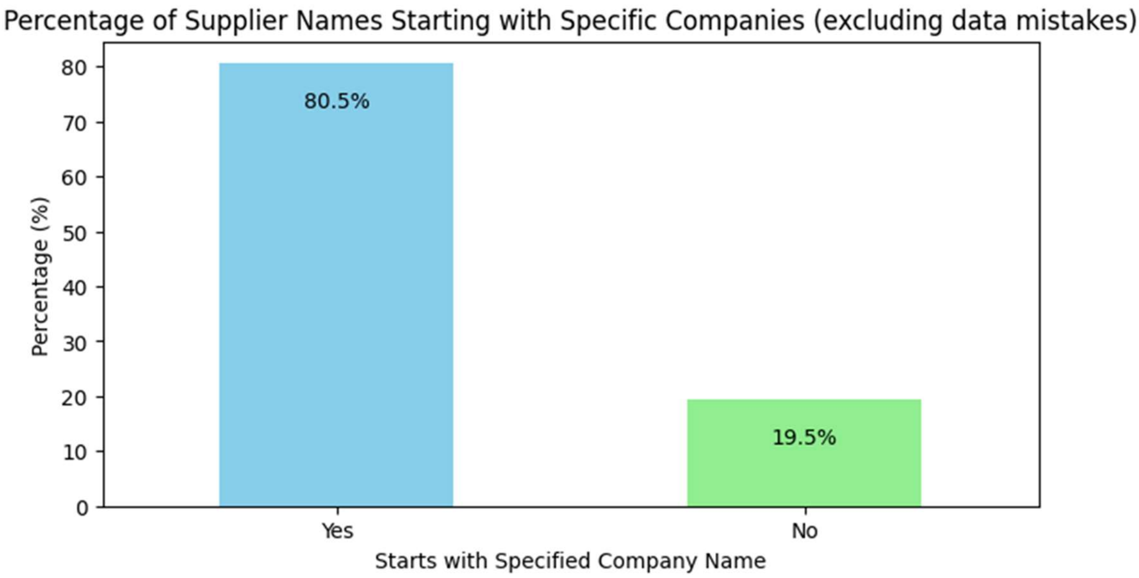


Figure 11. Percentage of Supplier name strings starting with the company name.

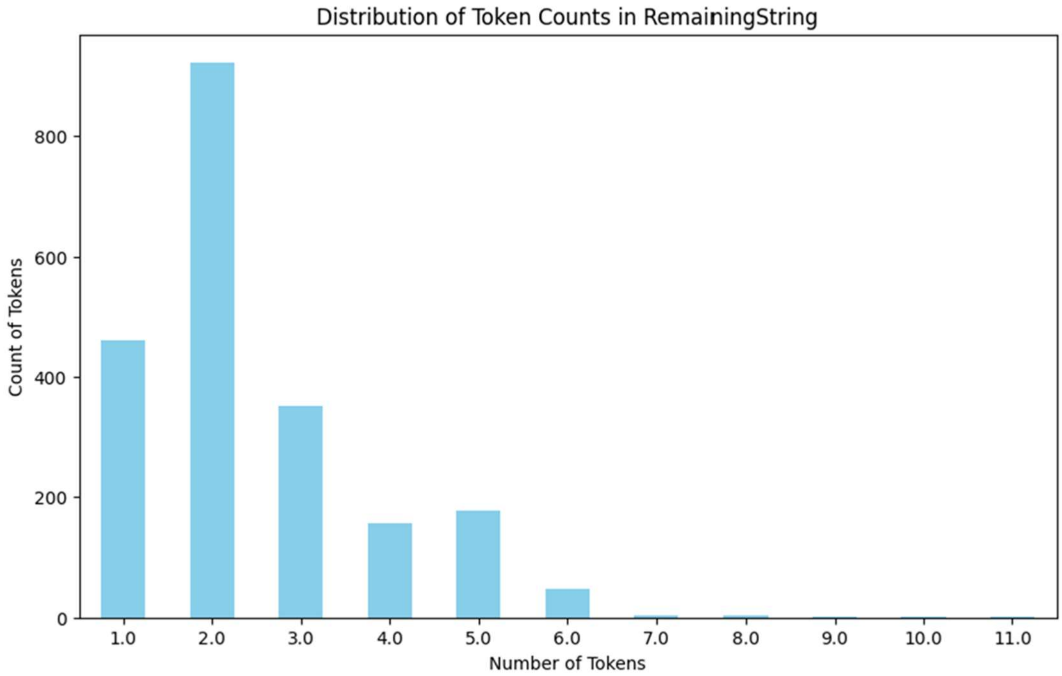


Figure 12. Token counts in remaining strings.

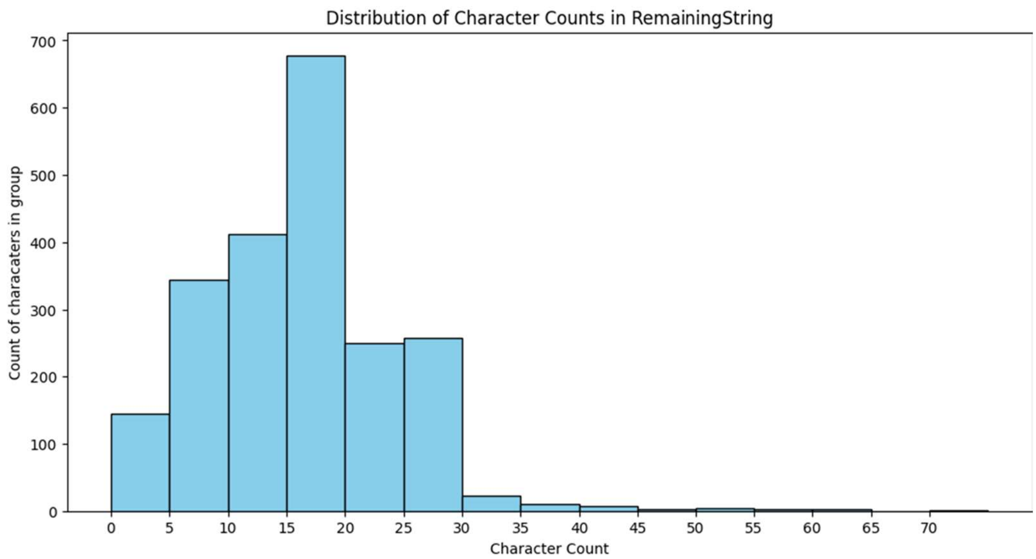


Figure 13. Character counts of remaining strings.

3.4 Summary of explorative analysis

After generating key statistics, the focus of the explorative analysis was to understand better what kind of patterns lead to manual corrections and also what patterns exist within the supplier groups which the system has failed to group together. The only meaningful pattern appearing in both datasets, which could be tackled with rule-based approach, was location. This means that if a location is identified and removed from the supplier or supplier group text string, it will enhance the linking of new supplier entering to system with existing supplier groups and improves grouping together the groups which are already in the system. However, there exist cases where the location is part of the actual supplier name, such as ‘Canada Goose’ which should be taken into account if put in further processing. Other identified patterns, such as handling the word ‘group’ or company abbreviations in similar fashion within the algorithms would also increase the linking performance moderately. Other grouped attributes, such as if the remaining string has real words or not or if foreign language is detected in the string, cannot be directly removed but could be useful if other approaches for the matching are considered.

By far the most impactful pattern is the supplier name itself in the supplier’s name text string and the question is then how to dig that information out of the string. In the current system, this is done manually by using keywords and for the companies which used key words, the linking performance was substantially higher than for the ones which did not use them. This highlights the limitations of the deterministic approach to the company name linking, there will be lots of missed matches, i.e. false negatives if the name is not recognized. After clear patterns are removed, it is very laborious in trying to improve the linking performance even by small fraction with deterministic rules, as one needs to define each potential case one by one. The complexity of the rules also adds up easily, as one would also need to identify what are the tradeoffs of creating a new rule. As the strings are natural language based, removing something from one string may lead to removing part of the actual company name on another occasion. The character counts and tokens in the remaining strings revealed that typically the supplier’s name remaining text string – when the actual company name has been removed - is having over 5 characters and 1-3 tokens. If we assume that the difference between 2 company name remaining string entries is 1 token having 5 characters and that the character could be any of the English alphabet (26 letters), this leads to 11881376 potential combinations. Even though some of the remaining strings are words and different lemmatizations of words, the number of potential combinations is still huge. In the next chapters we will examine if LLMs could be used to identify the company names from the text strings.

4 INFORMATION EXTRACTION USING LARGE LANGUAGE MODELS

4.1 Large language models in information extraction

Information extraction (IE) is an important area in the natural language processing that converts plain text into a structured format. Typical IE tasks involve named entity recognition (NER), event extraction (EE) and relation extraction (RE). NER includes tasks such as Entity Identification and Entity Typing which in practice means that for example first a name 'John' is identified and then assigned a type of a 'Person'. EE includes tasks such as event detection, which aims to identify and classify the trigger word and type that most clearly represents the occurrence of an event. Typical event detection example would be to identify trigger word and type from text '9.5 magnitude earthquake struck Kouvola yesterday, causing widespread damage to buildings and roads. Emergency services were quick to respond, providing aid to the affected areas.' Here the word 'struck' or 'causing' would be trigger words whereas 'earthquake' and 'damage' would be the event types. RE involves tasks such as relation classification which aims to classify the relation type between two given entities. (Xu et al., 2023)

Different methods have been introduced to tackle the information extraction problem and many state-of-the-art methods rely on pre-trained language models (PLMs) such as BERT. Although performing well, the models are having few major drawbacks: they require large amounts of human annotated training data for the context specific problem, and they don't generalize well on unseen data. Large language models have shown that they have potential to overcome these shortcomings. Also, the known issue of LLMs' propensity for generating unconstrained outputs can be mitigated with careful prompt engineering. The other known LLM issue, the inability to guarantee expert or human level performance, would require some human intervention depending on the problem and desired performance level. (Brinkmann et al., 2024; Goel et al., 2023)

4.2 Learning paradigms and prompt engineering in LLMs

Prompt engineering is relatively new discipline for developing and optimizing large-language model prompts for various applications and research topics. It helps to understand the capabilities and limitations of large language models. The area of prompt engineering is evolving all the time, but the main most widely used techniques particularly in Named Entity Recognition are zero-shot prompt engineering, few-shot prompt engineering and various other mixture methods of these. The prompt methods are also known as learning paradigms. (*Prompt engineering guide*, 2024; Xu et al., 2023)

4.2.1 Zero-shot prompt engineering

In natural language processing, zero-shot prompting means that the LLM is asked to perform a task without any examples or training data. Typical best practices to improve the results of zero-shot prompting is to divide the prompt into two sections. System and User. The 'System' defines what role the prompt user wants the LLM to take in the request and the 'User' is referred to the actual task on hand. Below picture describes what this means in practice.

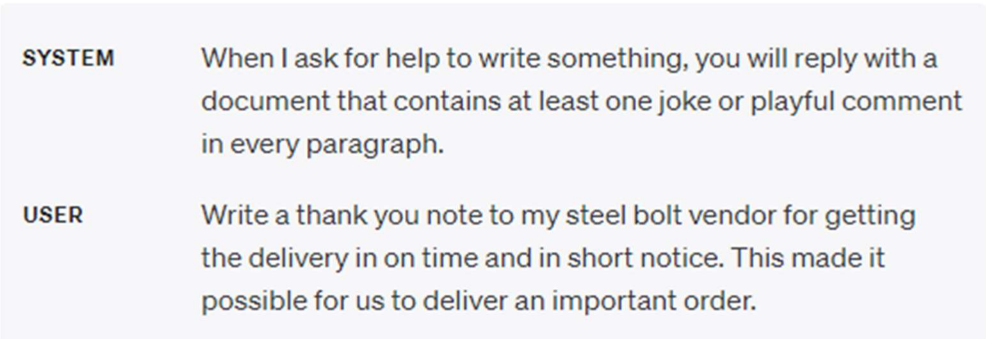


*Figure 14. Example system and user message in LLM prompt. (*OpenAI prompt engineering*, 2024).*

Another example of a prompt template when the zero-shot is used in attribute value extraction looks like below:

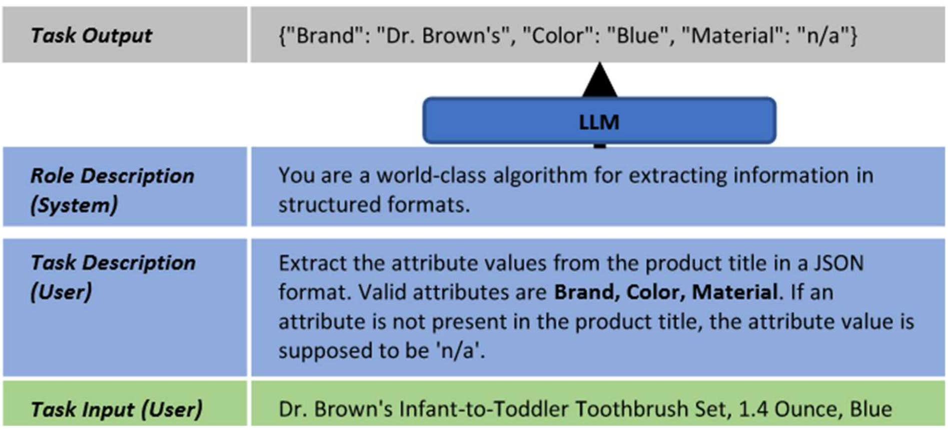


Figure 15. Example of zero-shot prompt in attribute value extraction. (Brinkmann et al., 2024).

Depending on the task at hand, there are various other tactics and strategies that can be used to improve the results of the prompting. OpenAI lists below 6 strategies which could be considered when optimizing the queries: 1. Write clear instructions, 2. Provide reference text 3. Split complex tasks into simpler subtasks. 4. Give the model time to ‘think’ 5. Use external tools 6. Test changes systematically. (*OpenAI prompt engineering*, 2024)

4.2.2 Few-shot prompt engineering

Few shot prompting means using LLM with limited number of labelled data. Compared to the zero-shot prompting, in few-shot prompting the user is giving the LLM example values what is wanted to be retrieved. Below picture illustrates example prompt template when the few-shot method is applied. In addition to the task and role descriptions, now the LLM is given also demonstration of the task input and output.

Role Description (System)	You are a world-class algorithm for extracting information in structured formats.
Task Description (User)	Extract the attribute values from the product title in a JSON format. Valid attributes are **Brand, Color, Material**. If an attribute is not present in the product title, the attribute value is supposed to be 'n/a'.
Demonstration – Task Input (User)	Quip Kids Electric Toothbrush Set - Electric toothbrush with multi-use cover (Green)
Demonstration – Task Output (Assistent)	{"Brand": "Quip", "Color": "Green", "Material": "n/a"}
Task Description (User)	Extract the attribute values from the product title in a JSON format. Valid attributes are **Brand, Color, Material**. If an attribute is not present in the product title, the attribute value is supposed to be 'n/a'.
Task Input (User)	Dr. Brown's Infant-to-Toddler Toothbrush Set, 1.4 Ounce, Blue

Figure 16. Example of few-shot prompt template in attribute value extraction. (Brinkmann et al., 2024)

5 EXPERIMENT SETUP

5.1 Data used

The experiment was divided into three samples which were processed by the OpenAI GPT API (Table 5).

Table 6. Test samples.

Sample	Details	Count of rows	Population size	Origin of the sample
Random sample before processing	Random sample of raw supplier data from ERP systems-typically from vendor master records.	300	Over 10M	Raw data before processed by the case company's entity resolution process
Known companies after processing	Sample of Fortune 500 companies identified in exploratory analysis having multiple supplier groups for same company. Sample consists of supplier group names within this group.	120	n/a	Output of the case company's entity resolution process
Medium companies after processing	Sample of companies not in the Fortune 500 group, but which have multiple supplier groups in system. for one real life company. Sample consists of supplier group names within this group.	120	n/a	Output of the case company's entity resolution process

For the random sample, statistical tests could be performed. The data consisted of raw data from customer ERPs, which means that the sample used is not impacted by the supplier entity resolution system used in the case company. The true population size is big, over 10 million rows, and the sample size of 300, 95 % confidence level was selected for the test. As the evaluation method of the test is not completely objective, the above parameters were deemed as optimal for the test as there would also be some bias coming from the evaluation method. Therefore, larger sample size with smaller margin of error would not balance out the efforts of working with larger sample.

The second and third samples were selected for the test from more practical reasons and from which truly random samples could not be obtained. Hence, statistical analysis for these samples could not be performed. However, as was revealed in the exploratory analysis, identifying the actual company names from text strings for these groups would still be highly beneficial from the system's end goal point of view, which is linking together supplier entities. Currently only the largest suppliers from spend point of view are reviewed manually and large number of suppliers outside the manual review process remain unnoticed. In addition, selecting companies which have multiple supplier groups in the system enabled us to analyze how well the LLM can extract the same company from different text strings. Also, having the output of the case company's current system as the starting point for the LLM, allowed us to investigate if process could be improved by using LLM model in the later stages of the case company's entity resolution process. Furthermore, having two different types of company groups in the samples (big and medium) enabled us to also see if there is any clear difference in LLM capability of extracting the company name between the two samples. Underlying assumption here was that as the large companies have probably more content online, there is probably more training data in the LLM model for these companies, which may make the probability of finding these companies better compared to smaller and not that known companies.

5.2 OpenAI API

OpenAI LLM was selected as the tool to perform the experiment because of two reasons: 1. In a study of (Brinkmann ym., 2024), which dealt with similar problem – extracting structured information from unstructured short texts – the OpenAI models performed in most cases the best. 2. There exist complete API documentation to run the experiment using Python programming language. Below is a picture of how the API was used. First the OpenAI library was installed using the normal pip install process. After importing the OpenAI library, a function was created to feed the company name lists to the API. API key was obtained from the OpenAI website and 20$ credit to use the models via API was bought. With the same API, it is possible to use different OpenAI models just by changing the model parameter. In addition to the model parameter, the other parameters that were tested with different combinations were the system_message, user_message and temperature.

```
import openai

def analyze_company_names(company_names):
    api_key = 'sk-[redacted]CJv0'
    client = openai.OpenAI(api_key=api_key)
    model = "gpt-3.5-turbo"
    results = []

    system_message = {
        "role": "system",
        "content": """You are a company AI assistant with expertise in natural language processing,
                    specifically in named entity recognition (NER) for identifying company, organization or
                    text strings.

                    Your task is to identify and extract main
                    company or organization entities from the given text string.

                    Text strings have often typos or intentional remarks such as xxx in front, ignore these.

                    """
    }

    for company_name in company_names:
        user_message = {
            "role": "user",
             "content": f"""Extract only the group, company or organization
             name from the company name string '{company_name}'.
             Remove words or tokens from the end or beginning if
             these are common words appearing behind many companies or they are not nee
             Return only one name found from the company string in list format."""
        }

        try:
            response = client.chat.completions.create(
                model=model,
                messages=[system_message, user_message],
                temperature=0
            )

            response_message = response.choices[0].message.content
        except Exception as e:
            response_message = f"Error processing {company_name}: {str(e)}"

        results.append((company_name, response_message))

    return results

list_of_companynames = RandomSuppliers
company_analysis_results = analyze_company_names(list_of_companynames)
```

Figure 17. Prompt template and function created for the API call.

5.3 Prompt template and parameters

Enhancing the prompt template was an iterative process. Best practices highlighted by the OpenAI (*OpenAI prompt engineering*, 2024) as well as users (Elwin, 2024) were tested. The best performance enhancements were obtained by applying a persona for the model, clearly defining, and stating the task, indicating in what format the response should be in as well as setting the temperature value to 0. Defining the persona means that the LLM is given a clear role what it should take. In this case the role was expert in Named Entity Recognition in identifying company related entities from text strings.

```
system_message = {
    "role": "system",
    "content": """You are a company AI assistant with expertise in natural language processing,
                specifically in named entity recognition (NER) for identifying company, organization or group name from
                text strings.
```

Figure 18. System message in used prompt.

Task definition was done to the system message giving further guidance for the LLM as well as to the task message which was sent to the LLM as many times as there were company text strings. One notion was the importance of stating what kind of responses should be generated as well as the specific format in which the response is expected in. In the first iterations, without the detailed messages, the tested models often returned multiple answers and even sentences in varying formats. Also json-format response output was tested as in other studies it has improved the response performance ((Brinkmann ym., 2024). However, in this case there was not any observed clear performance improvement, hence, list format for the responses was used.

```
                Your task is to identify and extract main
                company or organization entities from the given text string.

                Text strings have often typos or intentional remarks such as xxx in front, ignore these.

                """
    }

    for company_name in company_names:
        user_message = {
            "role": "user",
            "content": f"""Extract only the group, company or organization
            name from the company name string '{company_name}'.
            Remove words or tokens from the end or beginning if
            these are common words appearing behind many companies or they are not needed to identify the main company.
            Return only one name found from the company string in list format."""
        }
```

Figure 19. User message in used prompt template.

The temperature parameter defines how 'deterministic' the model should be in its responses. By design, the LLMs are probabilistic and not deterministic, but setting the temperature value low we want the model to output the most probable word or words. In other tasks, such as text or idea generation, setting the temperature higher may be good practice to get more variability to responses. For named entity recognition, best practice is to set the temperature to 0. (*GPT for work*, 2024; *Prompt Engineering Tips & Tricks for Named Entity Recognition (NER)*, 2024) When testing the models with different temperatures, the 0 was identified as the best parameter value in terms of quality and repetitiveness of the answers. It did not ensure fully similar responses between runs, but the variation between runs was minimized with the 0 temperature.

The chosen model to perform the final tests was gpt-3.5-turbo. In the first iterations also gpt-4 was tested but it did not improve the results in the task. As the pricing for the gpt-4 is much higher compared to the gpt-3.5-turbo, the experimenting continued using the gpt-3.5-turbo.

5.4 Evaluation method and tests

For the random sample, binomial test was carried out as we were interested only in one sample and the sample could have two different categories: correct or not correct against human review. For the technical computing, scipy.stats python library was used. (*Binom test scipy*, 2024). The 0 hypothesis of the test was based on the subjective idea of how good the model should be to use it in practice; therefore 90 % threshold was selected. The Null and Alternative hypotheses were formulated as:

Null hypothesis H(0): The Gpt-3.5-turbo model performance is 90 % or less compared to skilled human in finding company names from text strings

Alternative hypothesis H(A): The Gpt-3.5-turbo model performance is greater than 90 % compared to skilled human in finding company names from text strings.

The evaluation method for the tests required subjective reasoning as there is no 'truth-sets' available for the task. For all the samples, the LLM's capability in recognizing and extracting the company name from the unstructured text strings was compared to skilled human. In addition, a qualitative metric was monitored to identify what kind of output the model is giving when the answer differs from the best possible solution. This was considered important from practical reasons, as it is important to know what kind errors could be expected and because LLMs in general have been identified to hallucinate answers (Ma ym., 2023), which could severely undermine the automatic use of the responses. The 'good' answer was relating to only what could be interpreted from the text string. For example, if the name 'Google Inc' was shown in the text string, recognizing, and extracting 'Google' was considered correct and the real group name 'Alphabet' was not required. The human review was also strict in a sense that to get correct answer, the model was required to output only the part of the company name from the text string how it could be identified by using as small amount of characters and tokens as possible. For example, if the input company name text string was 'Exxon Mobile Medicare Supplement', recognizing and extracting 'Exxon Mobile' was considered incorrect as only 'Exxon' would have been sufficient.

Table 7. Evaluation methods and statistical tests for the samples.

	Random sample	**Known companies**	**Medium companies**
Statistical test	Binomial test	n/a	n/a
Quantitative Evaluation method	Comparison against human reviewed best scenario. Quantifying correct vs. not correct.		
Qualitative evaluation method	Human review, what kind of output is model generating when response is not correct		

6 RESULTS

6.1 Quantitative findings from random sample results

The first measurement of the random sample was to check how many of the model responses are correct. Correct below means that the model had outputted same response as skilled human would do. Out of the 300 text strings, 281 were correct and 19 were not correct totalling the correct % to 93,67.

Table 8. Performance of the gpt-3.5-turbo model for random sample.

Correct	93,67 %
Not correct	6,33 %

The 0-hypothesis stated that 'The Gpt-3.5-turbo model performance is 90 % or less compared to skilled human in finding company names from text strings'. With 95 % confidence level, the 0 hypothesis is rejected, because the p-value ends up under 0,05.

```
result = binomtest(281, n=300,p=0.9 , alternative= 'greater')
print(result.pvalue)
✓ 0.1s
0.017118132722922835
```

Figure 20. Result of the scipy.binomtest p-value.

This means that the alternative hypothesis is valid: The Gpt-3.5-turbo model performance is greater than 90 % compared to skilled human in finding company names from text strings. The result means that if the experiment would be re-run x-times with random sample, 95 % of the runs would end up with model performance inside the upper and lower confidence interval (Figure 21).

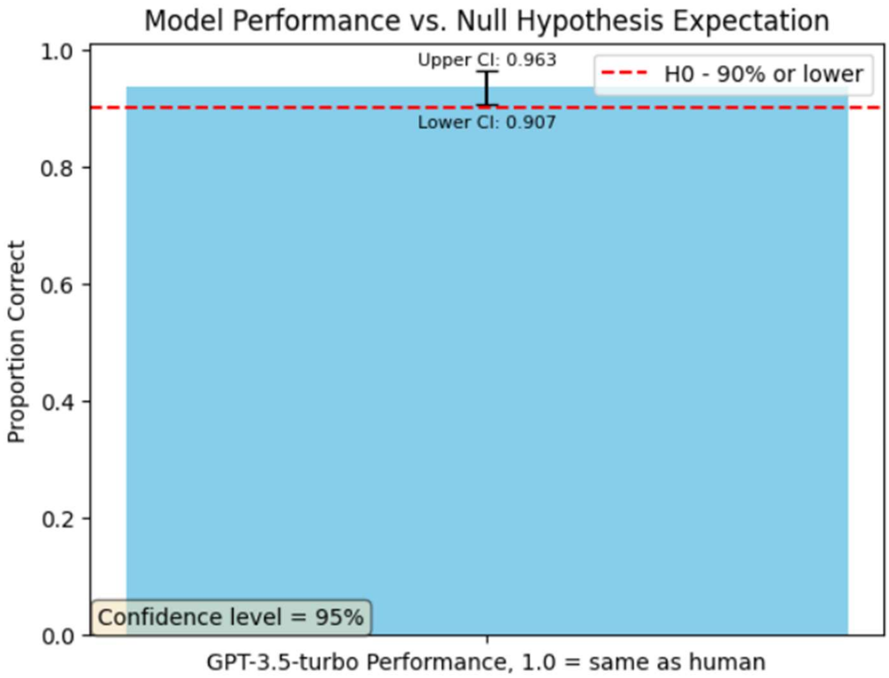


Figure 21. Model performance vs. null hypothesis.

6.2 Qualitative findings from the random sample results

Below sample of the correctly found company names (total of 281/300) show that the model is relatively good at finding the relevant company name from the text strings and on the other hand, not removing too many tokens or characters. For example, the company abbreviations as well as unnecessary extra characters in the end or beginning of strings are removed efficiently. The first column indicates the input name to the model and the second column is the model's response.

InputCompanyName	OutputGroupName	Human validation correct
PANERA BREAD [redacted]	["PANERA BREAD"]	x
HRG BELGI ([redacted]	['HRG BELGI']	x
SUBHASH INSTRUMENTS - [redacted]	['SUBHASH INSTRUMENTS']	x
HRG2 [redacted]	['HRG']	x
VAPIANO AMSTELSTRAAT	['VAPIANO']	x
AUSTBULK SHIPPING PTE LTD	['AUSTBULK SHIPPING']	x
Pošta Slovenije	['Pošta Slovenije']	x
Yong Lee Enterprise Sdn Bhd	['Yong Lee Enterprise']	x
GOGGINS ENTERPRISES 1 INC	['GOGGINS ENTERPRISES']	x
JOSEPH BISKUPIAK	['JOSEPH BISKUPIAK']	x
[redacted] UNAND-UTK DANA KELOLA SCIEN	['DANA KELOLA SCIEN']	x
NIUTTA RICCARDO	['NIUTTA RICCARDO']	x
ETT Elektrotechnik Team GmbH	['ETT Elektrotechnik Team']	x
LEROY GUY SA	['LEROY GUY']	x

Figure 22. Example of correct model outputs.

However, there were also cases where the model failed to give correct responses. The 19/300 incorrect cases are listed below.

InputCompanyName	OutputGroupName	Human validation not correct
RESTAURANT AIKAWA	['AIKAWA']	x
pia1	["pia"]	x
JOHN LEO CAPLAN	['CAPLAN']	x
Univer Sweden AB	['Univer Sweden']	x
SLICE OF STAMFORD	['STAMFORD']	x
SQ SQ LA MEDITERRANE	['LA MEDITERRANE']	x
PIECHOTA'S	['PIECHOTA']	x
TORONTO AUTO DEPOT LTD DBA: TORONTO AUTO DEPOT	['TORONTO AUTO DEPOT LTD']	x
FARM SFERLAZZO DR MARCO	['SFERLAZZO']	x
TOODALEESHIKDANG	['TOODALEE']	x
PRIMUS AUTOMOTIVE FINANCIAL SERVICES CANADA INC	['PRIMUS AUTOMOTIVE FINANCIAL SERVICES CANADA']	x
Home to Baiyun (CN) airport	['Baiyun']	x
SYSTEMES CLIMATIQUES SERVICE	['SYSTEMES CLIMATIQUES']	x
In-Albon-Urfer/Giacinto Oder Verena	['In-Albon-Urfer', 'Giacinto', 'Verena']	x
ZZ TRAST dd	['ZZ TRAST']	x
DIAMOND Piotr Dulas	['DIAMOND']	x
CHOCOLATE BAR GARDEN	['CHOCOLATE BAR']	x
MAGDALENA WĄSOWSKA ZAKŁAD/PIEKARNICZO-CUKIERNIC	['MAGDALENA WĄSOWSKA']	x
DREAM MAKERS AUTOMOTIVE	['DREAM MAKERS']	x

Figure 23. All incorrect model outputs.

Some of the cases are clearly wrong and misleading, such as ‘Restaurant Aikawa’ → ‘Aikawa’, ‘Dream Makers Automotive’ → ‘Dream Makers’, ‘Home to Baiyn (CN) Airport’ → ‘Baiyen’ and ‘Slice of Stamford’ → ‘Stamford’. Others on the other hand are not misleading, but the strings still contain unnecessary information even though some cleaning has taken place. Examples for these are: ‘Univer Sweden’, ‘Primus Automotive Financial Services Canada’ and ‘Toronto Auto Depot LTD’. On the positive side, the incorrect responses do not indicate any kind of hallucination from the model.

6.3 Quantitative findings from the Known and Medium companies’ results

The dataset formulation for the Known and Medium samples was different compared to the random sample. In these samples the input data was the output of case company’s current system, making the samples harder for the model at least in theory, because the easy cases have already been cleaned by the case company’s current process. The results against the human validation for Known Companies and Medium Companies samples are presented below. The Known Companies sample had 72 % correct responses whereas the Medium companies had 83 % correct answers from the gpt-3.5-turbo model.

Table 9. Performance of the gpt-3.5-turbo model for Known companies sample.

Correct	71,67 %
Not correct	28,33 %

Table 10. Performance of the gpt-3.5-turbo model for Medium companies sample.

Correct	83,33 %
Not correct	16,67 %

Another way to view the results was to check what should be the perfect number of groups inside the samples and compare that against the model output. For the Known Companies sample, the model was able to reduce the number of groups from 120 to 34 while 9 was the perfect amount. With the Medium Companies sample, the same figures were 120 to 42 while 23 distinct groups would have been the perfect result.

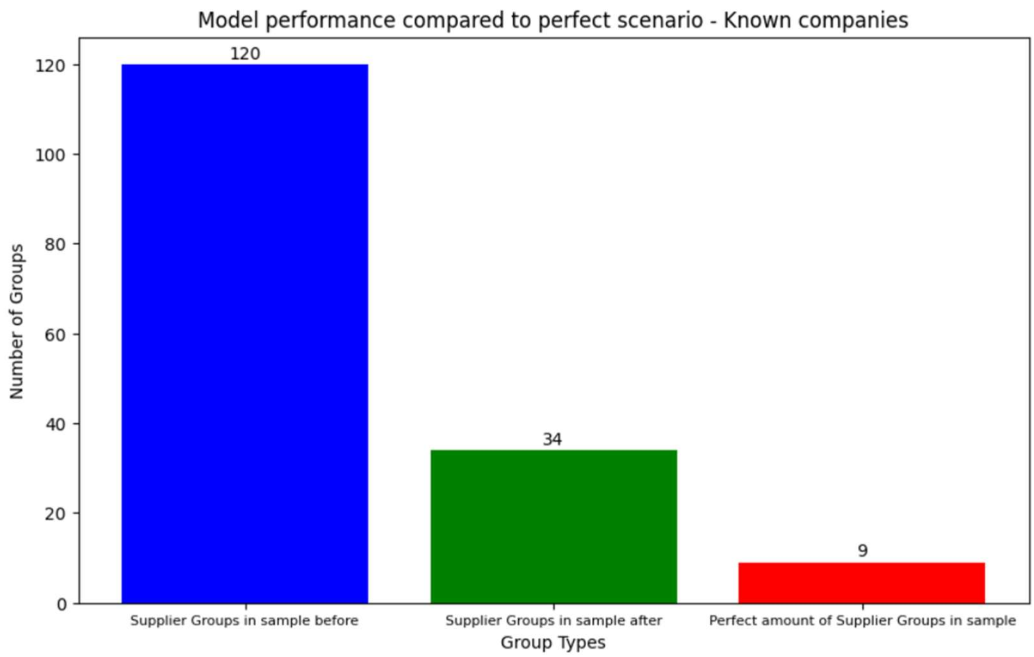


Figure 24. Model performance against perfect scenario for Known Companies sample.

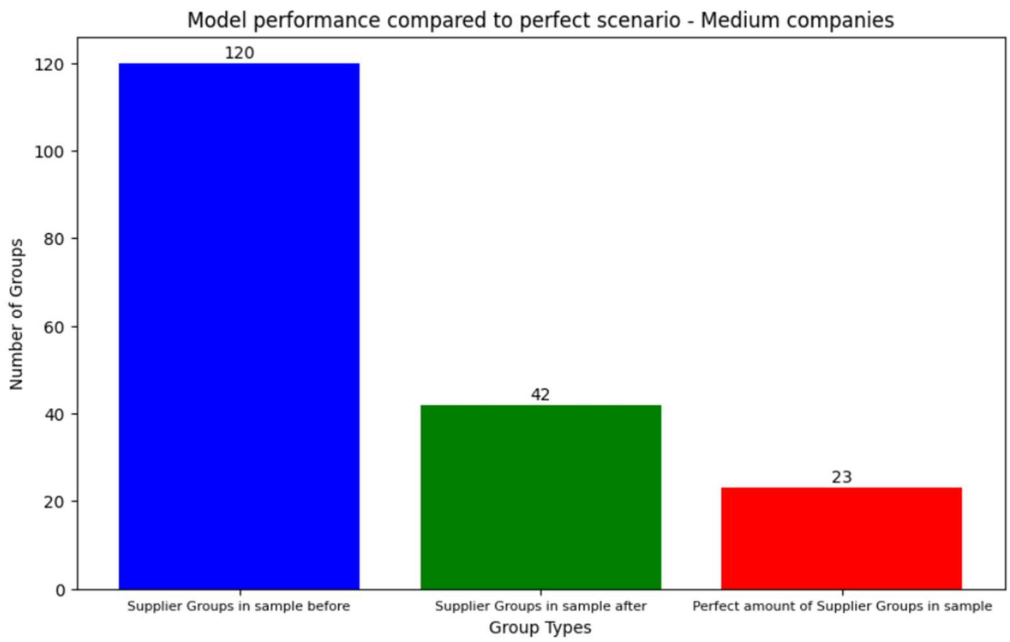


Figure 25. Model performance against perfect scenario for Medium companies sample.

6.4 Qualitative findings from Known and Medium companies' results

For Known Companies sample, the model was able to find some of the more difficult cases also, where the string does not start with the company name, such as 'Mvp Select Care Verizon' → 'Verizon' and 'Blocked Use Caterpillar' → 'Caterpillar'.

InputCompanyName	OutputGroupName	Human validation correct
Verizon Italia	["Verizon"]	x
Ecolab Pest Elimination Division	['Ecolab']	x
Henkel Ecolab Institutional	['Henkel', 'Ecolab']	x
Verizon Lehigh Valley	['Verizon']	x
Manpower Lorient	['Manpower']	x
Verizon Maryland	['Verizon']	x
Manpower Uitzendbureau	["Manpower"]	x
Verizon Engineering	["Verizon"]	x
Worrell Exxon	['Exxon']	x
Mvp Select Care Verizon	['Verizon']	x
Verizon Internet Services	['Verizon']	x
Caterpillar European Hose	['Caterpillar']	x
John Deere Forestry	['John Deere']	x
Verizon New Zealand	['Verizon']	x
Ecolab Nd	[Ecolab]	x
Blocked Use Caterpillar	['Caterpillar']	x
Manpower Secme Ve Yerlestirme	['Manpower']	x
Exxon Medical Plan	['Exxon']	x
Manpower Dorlyl	['Manpower']	x

Figure 26. Example of correct model outputs for Known companies sample test.

Many of the incorrect responses (Figure 30.) were due to the strict rule of handling the company 'Exxon', where the perfect answer was to output only 'Exxon' not 'Exxonmobile' or 'Exxon Mobil'. For almost all the companies which were categorized as incorrect, substantial clean-up had still happened without removing vital tokens or characters. Also, there was no signs of fully misleading response outputs or signs of response hallucination.

InputCompanyName	OutputGroupName	Human validation not correct
Exxon Mobil Chemical Films Europe	['Exxon Mobil']	x
Exxon Mobil Chemical Central	['Exxon Mobil Chemical']	x
Exxon Mobile Medicare Supplement	['Exxon Mobile']	x
Verizon Wireless La	['Verizon Wireless']	x
Exxonmobile Chemical	['Exxonmobile']	x
Henkel Ecolab	['Henkel']	x
Exxonmobil Chemical Mediterranea	['Exxonmobil']	x
Verizon Wireless Ca	['Verizon Wireless']	x
Exxonmobil Oil Wilcox & Flegel	['Exxonmobil']	x
Exxonmobil Cat Outside	['Exxonmobil']	x
Exxonmobil Research & Engineering	['Exxonmobil']	x
Exxonmobil Chemical Holland	['Exxonmobil Chemical']	x
Verizon Wireless Bellevue	['Verizon Wireless']	x
Exxonmobil Middle East & Africa	["Exxonmobil"]	x
Manpower Group Japan	["Manpower Group"]	x
Manpower Services	['Manpower Services']	x
Manpower Monitoring	["Manpower Monitoring"]	x
Exxonmobil Production	['Exxonmobil']	x
Exxonmobil Chemical Films	['Exxonmobil']	x
Exxonmobil Lubricants &	['Exxonmobil']	x
Blocked Manpower Staffing Services	['Blocked Manpower Staffing Services'	x
Exxon Mobile Medical Plan	['Exxon Mobile']	x
Verizon Wireless Services	['Verizon Wireless']	x
Verizon Wireless Messaging Svc	['Verizon Wireless']	x
Aecom Australia & Arcadis	['Aecom Australia', 'Arcadis']	x
Verizon Communications Technology	['Verizon Communications']	x
Aecom Technical Services Of	['Aecom Technical Services']	x
Deutsche Exxon Chemical	['Deutsche Exxon']	x
Verizon Wirless Sd	['Verizon Wireless']	x
Verizon Communications	['Verizon Communications']	x
Rainfocvmware Vmworld	['Rainfocvmware']	x
Exxonmobil Lubricants & Specialties	['Exxonmobil']	x
Exxon Mobil Chemical	['Exxon Mobil']	x
Verizon Digital Media Services	['Verizon Digital Media Services']	x

Figure 27. All incorrect model outputs for Known companies sample test.

The Medium Companies sample followed similar patterns as the Known Companies. The model was quite good in finding the relevant tokens and characters from the text strings, although the result was not perfect.

InputCompanyName	OutputGroupName	human validation correct
Daylight Donuts Cl	["Daylight Donuts"]	x
P.G. Daylight Donuts	['Daylight Donuts']	x
Daylight Donuts Of Pe	['Daylight Donuts']	x
Daylight Donuts Muskogee	['Daylight Donuts']	x
Daylight Donuts of Saratoga Springs	['Daylight Donuts']	x
Daylight Donuts Sandy	['Daylight Donuts']	x
Universal Express Aalst	['Universal Express']	x
Universal Express Zeebrugge	['Universal Express']	x
Universal Express Star	['Universal Express']	x
Universal Express Magazun Cbg	['Universal Express']	x
Invalid Universal Express	['Universal Express']	x
Tcc Telecomconsult	['Tcc Telecomconsult']	x
Comconsult Suppan	['Comconsult']	x
Comconsult Beratung Und Planung	['Comconsult']	x
Comconsult Kommunikations	['Comconsult']	x
Comconsult Beratung	['Comconsult']	x
Triangle Engineering Of Arkansas	['Triangle Engineering']	x
Triangle Engineering Of A	['Triangle Engineering']	x
Shoppartners Beheer	['Shoppartners']	x
Shoppartners	['Shoppartners']	x
Shoppartners Nl	['Shoppartners']	x
Donauchemieag	['Donauchemie']	x

Figure 28. Example of correct model outputs for Medium companies sample test.

The full list of mistakes for the Medium sample (20 in total) are listed below. There was two cases where the model output is clearly misleading: 'Airport (Hudson News)' → 'Airport' and 'Galaxy Motorcars Dba Galaxy Motors' → 'Galaxy Motorcars'. Again, in the medium sample there was no signs of response hallucination as all the answers can be found also in the input text string.

InputCompanyName	OutputGroupName	human validation not corre
Boling Daylight Donuts	['Boling Daylight Donuts']	x
Universal Express Services O	['Universal Express Services']	x
Universal Express Logistic	['Universal Express Logistic']	x
Universal Express Distribution	['Universal Express Distribution']	x
Atl Universal Express	['Atl Universal Express']	x
Universal Express Relocations	['Universal Express Relocations']	x
Comconsult Akademie	["Comconsult Akademie"]	x
Comconsult Research	['Comconsult Research']	x
Donauchem Vegyianyag Kereskedelmi K	['Donauchem Vegyianyag']	x
Airport (Hudson News)	['Airport']	x
Suku Druck- Und Temperaturmeßtechnik	['Suku Druck- Und Temperaturmeßtechnik']	x
SUKU Druck- und Temperatur-	['SUKU']	x
California State University System	['California State University System']	x
California State University San Marcos Foundation	['California State University San Marcos Founda	x
Xxxhuron Consulting Services	['Huron Consulting Services']	x
Huron Consulting Group	['Huron Consulting Group']	x
Vishay Precision Group	['Vishay Precision Group']	x
Vishay Precision Group Canada	['Vishay Precision Group']	x
Galaxy Motorcars Dba Galaxy Motors	['Galaxy Motorcars']	x
Emerson Process Managements	["Emerson Process Managements"]	x

Figure 29. All incorrect outputs for the Medium companies sample test.

Surprisingly, there did not appear to be difference in performance in terms of whether the company is very big 'Fortune 500' or a smaller company. Even thought it was not possible to statistically test the assumption, the results indicate that the gpt-3.5-turbo's capability in finding company entities from text strings does not require the entity to be very large.

7 CONCLUSIONS AND DISCUSSION

The thesis started by describing the current entity resolution process of the case company as well as identifying the patterns in supplier text strings which lead to incorrect or suboptimal supplier groupings. Case company's supplier entity resolution process is by design deterministic, meaning that in order to link two or more text strings (suppliers) together, they need to agree on all common attributes used by the algorithm in the matching phase. This approach has its pros and cons. Benefit being the minimized number of false matches at the expense of false non-matches. When the incorrect or suboptimal groupings were examined further, by using a dataset of manually corrected entries and a sample of already processed output of large Fortune 500 supplier groups, certain often appearing text patterns emerged. The easier patterns from the perspective of logic changes in current system were location and the way the algorithms are handling company endings and abbreviations, such as word 'group'. However, the total impact of aligning and/or removing these tokens could be considered moderate or even small. By far the most impactful pattern in the supplier text strings is the company name itself and identifying and extracting the name would have significant performance improvement. Not surprisingly, the current use of manually given keywords indicates that the supplier groups which are attached with key words and processed by the keyword algorithm, perform better in terms of grouping relevant suppliers together. It can also be argued how effort efficient it is in trying to improve the system performance using rules after the clear patterns have been cleaned. The Fortune 500 supplier group text analysis showed that once the company name is removed from a string, it still has well over 5 characters. As the supplier names are entered in free text fields in source systems, the potential combinations preceding or following a company name is huge and hence, makes the rule creation cumbersome at best.

As there has been big improvement in recent years on LLM capabilities in text understanding and generation, the second part of the thesis focused on experimenting with the OpenAI gpt-3.5-turbo model how well it could recognize and extract company names from given text strings. The experiment was divided into three samples to test different aspects of model performance. First sample consisted of randomly selected raw data before processed by the case company's entity resolution system. The second and third samples were the outputs of the case company's process. For all the samples, the model performance was compared against skilled human. For the random sample, statistical binomial test was performed with a Null-hypothesis stating that The Gpt-3.5-turbo model performance is 90 % or less compared to skilled human in finding company names from text strings.

The test showed performance of 93,67 % with p-value of 0,017 and hence, null hypothesis was rejected with 95 % confidence level. The Known and Medium companies samples had performance of 71 % and 83 % respectively. The other quantified measurement for the Known and Medium companies sample was how much are the amounts of groups reduced by using the model. For the Known companies, the groups reduced from 120 to 34 while 9 was the perfect result. For Medium company sample, the same numbers were 120, 42 and 23. From qualitative perspective, also the model responses, and specifically the incorrect responses were analysed. The model did not show any signs of hallucination in the responses. The incorrect responses were misleading in couple of cases, meaning that the model had removed too many characters. By far the largest number of incorrect responses were due to the model not cleaning all the characters and tokens it should've.

7.1 Practical usability of the results

The performance level of the model was relatively good and suggests that the gpt-3.5-turbo or other similar LLM model should be considered when aiming to improve the case company's entity resolution system of linking supplier entities. Specifically, as the system is currently deterministically oriented, LLM would give a more probabilistic tool in identifying and extracting the company names from text strings. In addition, in the cases where the model did fail to give correct output, the output was not random word, but usually extra token or characters existing in the input text string, meaning that the impact of the wrong output is mitigated. Only in a few cases the model output too short or misleading text string. If applied in practice, the output of the model could be used to identify companies that should belong to the same supplier groups within existing groups. Also, the model output could be used in the cleaning phase of the process alongside the existing algorithms. Below Figure 30. illustrates the potential use cases. The use case 1, identifying duplicates within existing groups would be more straightforward to implement and would not require changes to existing system as the LLM model output could be used in cleaning the existing system by identifying duplicate groups. The use case 2 would require changes to existing system, as the output of the LLM should be stored before the keyword algorithm could check potential matches against LLM proposed company names. The use case 2 could potentially be applied as an automated process for the cases where the impact of incorrect grouping is low. These would be mainly cases where the supplier is small from spend metric point of view, meaning that the supplier is not that important for the customer. Below cumulative distribution function (Figure 31) highlights that the amount of suppliers which have substantial amount of spend, and which are more important to manage from customers' costs point of view, is small compared to the total amount of suppliers. Typically, companies adhere to a Pareto principle in their purchasing patterns, where a majority of their procurement spending is concentrated with just 10% or fewer of their total suppliers. Consequently, automating the matching process for the numerous smaller suppliers could prove beneficial, despite potential trade-offs in quality. Conversely, accurate groupings are crucial for key suppliers. Therefore, integrating human validation into the automation process for LLM-based company name identification is essential, as the current LLM performance does not yet match the level of a skilled human.

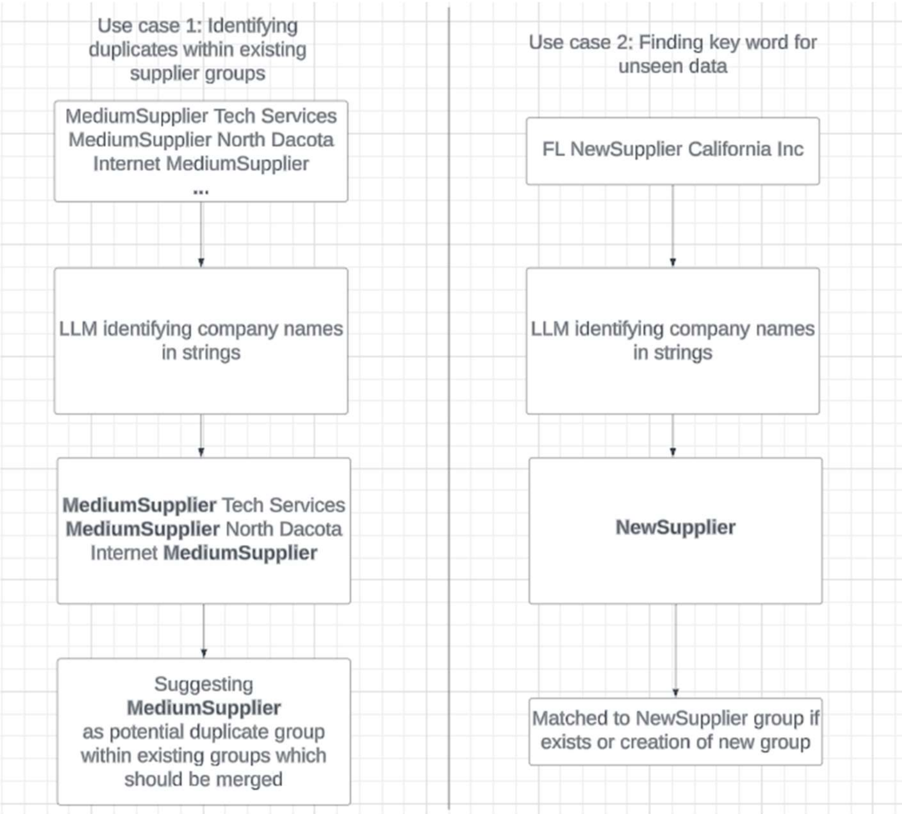


Figure 30. Potential use cases.

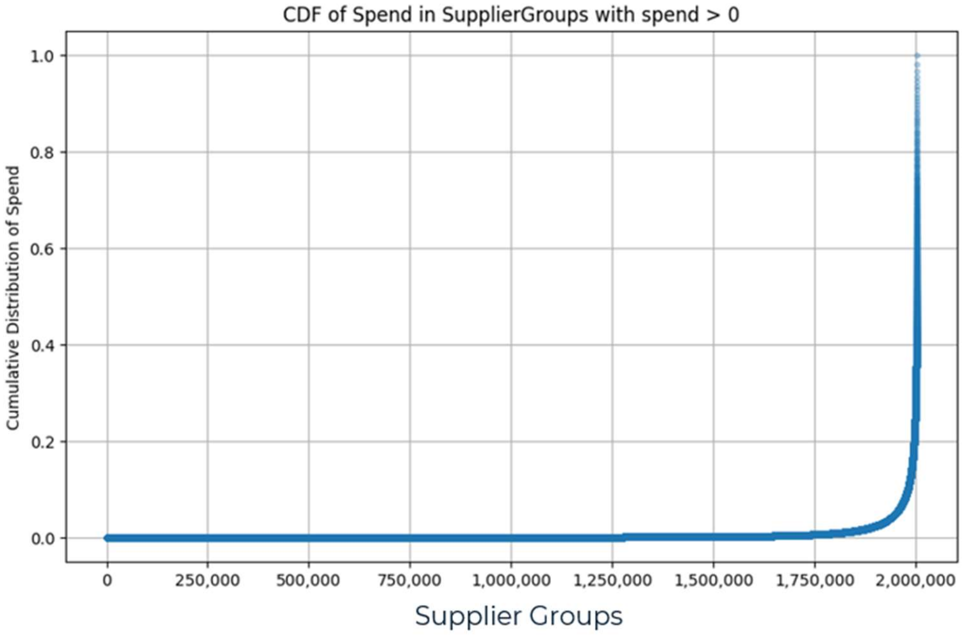


Figure 31. Cumulative distribution of spend for supplier groups in system.

7.2 Future research

The study was conducted using zero-shot prompt engineering techniques in aiming to find the relevant information from the short text strings. It would be interesting to test the setup with other learning paradigms, such as few-shot or mixture methods to see if they would improve the quality of the results.

REFERENCES

Binette, O., & Steorts, R. C. (2022). *(Almost) All of Entity Resolution* (arXiv:2008.04443). arXiv. http://arxiv.org/abs/2008.04443

Binom test scipy. (2024, 5). https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binomtest.html#scipy.stats.binomtest

Brinkmann, A., Shraga, R., & Bizer, C. (2024). *Product Attribute Value Extraction using Large Language Models* (arXiv:2310.12537). arXiv. http://arxiv.org/abs/2310.12537

Christen, P. (2019). Data Linkage: The Big Picture. *Harvard Data Science Review*, *1*(2). https://doi.org/10.1162/99608f92.84deb5c4

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2020). *End-to-End Entity Resolution for Big Data: A Survey* (arXiv:1905.06397). arXiv. http://arxiv.org/abs/1905.06397

Elwin, M. K. (2024, huhtikuuta 10). Prompt Engineering Tips & Tricks for Named Entity Recognition (NER). *Medium*. https://medium.com/@marcuskelwin/prompt-engineering-tips-tricks-for-named-entity-recognition-ner-090d6e38afc3

Goel, A., Gueta, A., Gilon, O., Liu, C., Erell, S., Nguyen, L. H., Hao, X., Jaber, B., Reddy, S., Kartha, R., Steiner, J., Laish, I., & Feder, A. (2023). *LLMs Accelerate Annotation for Medical Information Extraction* (arXiv:2312.02296). arXiv. http://arxiv.org/abs/2312.02296

GPT for work. (2024, heinäkuuta 5). https://gptforwork.com/guides/openai-gpt3-temperature

Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing*, *28*(3), 539–547. https://doi.org/10.1007/s11222-017-9746-6

Ma, Y., Cao, Y., Hong, Y., & Sun, A. (2023). *Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!* (arXiv:2303.08559). arXiv. http://arxiv.org/abs/2303.08559

OpenAI prompt engineering. (2024, toukokuuta 5).

Prompt engineering guide. (2024, toukokuuta 5). https://www.promptingguide.ai/techniques/zeroshot

Prompt Engineering Tips & Tricks for Named Entity Recognition (NER). (2024, toukokuuta 5). https://medium.com/@marcuskelwin/prompt-engineering-tips-tricks-for-named-entity-recognition-ner-090d6e38afc3

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., & Chen, E. (2023). *Large Language Models for Generative Information Extraction: A Survey* (arXiv:2312.17617). arXiv. http://arxiv.org/abs/2312.17617