

LLM-based Approaches for Automatic Ticket Assignment: A Real-world Italian Application

Nicola Arici^{1,2,*}, Luca Putelli¹, Alfonso E. Gerevini¹, Luca Sigalini² and Ivan Serina¹

¹Department of Information Engineering, University of Brescia, Via Branze 38, Brescia, Italy

²Mega Italia Media, Via Roncadelle 70A, Castel Mella, Italy

Abstract

IT service providers need to take care of errors, malfunctions, customizations and other issues every day. This is usually done through tickets: brief reports that describe a technical issue or a specific request sent by the users of the service. Tickets are often read by one or more human employees and then assigned to technicians or programmers in order to solve the raised issue. However, the increasing volume of such requests is leading the way to the automatization of this task. Since these tickets are written in natural language, in this paper we aim to exploit the new powerful pre-trained Large Language Model (LLM) GPT-4 and its knowledge in order to understand the problem described in the tickets and to assign them to the right employee. In particular, we focus our work on how to formulate the request to the LLM, which information is needed and the performance of different zero-shot learning, few-shot learning and ensemble learning approaches. Our study is based on a real-world ticket dataset provided by an Italian company which supplies IT solutions for creating and managing online courses.

Keywords

Automatic Ticket Assignment, Large Language Models, Prompt Engineering, Text Classification

1. Introduction

Modern companies which supply IT solutions not only have to provide an effective software environment, but they also need to maintain it during the software lifecycle, to fix errors and malfunctions, to introduce new functionalities and satisfy the requests submitted by the users. This task is usually done by programmers specialized in maintenance tasks which need to take care of new issues every day.

Such issues are usually submitted through ticketing systems. In these systems, the users can write a brief report that describes a problem they encountered, or a specific service they need. These reports, typically called *tickets*, are then distributed among the maintenance specialists which have to satisfy the users' requests. However, in large companies which provide complex IT solutions or more than one product, different employees devoted to the maintenance can have different expertise. Therefore, there is the need to assign a ticket to the right person, i.e. an employee who has the necessary technical skills to solve the raised issue.


NL4AI 2023: Seventh Workshop on Natural Language for Artificial Intelligence, November 6-7th, 2023, Rome, Italy [1]

*Corresponding author.

✉ nicola.arici@unibs.it (N. Arici); luca.putelli@unibs.it (L. Putelli); alfonso.gerevini@unibs.it (A. E. Gerevini); luca.sigalini@megaitaliamedia.it (L. Sigalini); ivan.serina@unibs.it (I. Serina)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In order to do that, typically one or more human employees have to read the ticket, understand the request and assign it to a technician. Since this task is quite time consuming, bigger companies are starting to implement automatic solutions. Although these solutions can be based on ad hoc algorithms [2, 3] or on fine-tuning generic pre-trained language models [4] such as BERT [5], they would require a considerable amount of training data and an expensive effort (by programmers and machine learning specialists) to implement such models. On the other hand, the outstanding results obtained by pre-trained large language models (LLMs) as few-shot learners (i.e. with a minimal number of training examples) [6, 7, 8] could make automatic ticket assignment available to many companies even without any particular effort.

In order to verify whether that is achievable, in this work we investigate how these models can be applied to a real-world case scenario: the assignment of the tickets received by the Italian company Mega Italia Media¹, which provides IT solutions in the e-learning sector for the occupational safety. In particular, in 2011 they released the DynDevice² Learning Management System (LMS), facilitating companies in standard corporate training, allowing them to create specific courses, managing final exams [9, 10] (providing also the related certificates if the exam has been passed) and the interaction with the users [11]. The company receives many tickets related to this platform, which has to be maintained and updated constantly in order to satisfy the users' needs. Using these tickets, we verify the performance of OpenAI GPT-4 [12], a state-of-the-art pre-trained LLM based on the Transformer architecture [13], for this task.

However, it has been noted that the performance of such models can significantly vary depending on how the task requested is formulated or, in more technical terms, which *prompt* has been used [14, 15]. Therefore, we study different configurations and prompts into which more or less information is available to the LLM and in terms of how many examples we provide. We compare these results with a baseline into which a BERT model is fine-tuned on this task with 1000 labeled tickets.

The rest of the paper is organized as follows. In Section 2, we provide the background and an overview of the state-of-the-art and the related works. In Section 3, we describe the dataset of our application. In Section 4, we describe our approaches for solving the automatic ticket assignment task, which are evaluated and discussed in Section 5. Finally, in Section 6 we propose some conclusions and future developments. The code and the datasets can be found on GitHub³

2. Related work

In recent years, several researchers have approached the support ticket domain, solving problems such as ticket categorization [2, 4, 16], ticket assignment and ticket resolution [17]; our work falls in the second category.

In 2018, Uber, the famous private car transport company, proposed *COTA* [18] (Customer Obsession Ticket Assistant), a framework to take care of customer issues. They proposed two versions of their system: the first one combines several features, such as user information, trip information and ticket metadata, with a Random Forest algorithm for predicting the correct

¹<https://www.megaitaliamedia.com/en/>

²<https://www.dyndevice.com/en/>

³<https://github.com/nicolarici/AI-TS>

operator of each ticket; the second version leverages a Encoder-Combiner-Decoder approach, based on CNN and RNNs over different types of features (such as categorical, numerical, binary and text features) and a multi-classification layer.

A similar approach has been developed by DeLucia and Moore [19]; the authors implemented a Random Forest model fed with features created with latent Dirichlet allocation topic modeling, latent semantic analysis and Doc2Vec [20] starting from the ticket subject and message.

Han and Sun [21] proposed in 2020 *DeepRouting*, an intelligent system for assigning tickets to operators in an expert network. It contains two modules: one for text matching, based on a convolutional neural network trained over tri-grams derived by the ticket description, and one for graph matching, based on a Graph Convolutional Network fed with the experts graph.

With Feng et al. [22], in 2021 Apple developed its personal ticket assignment system, *TaDaa* (Ticket Assignment Deep learning Auto Advisor). This system is based on the state-of-the-art Transformer architecture, in particular a pretrained BERT model fine-tuned to solve two classification tasks. The model has two different classifiers: the first one to assign the ticket to one of the 3000 groups, and the second one to identify the expert (that belongs to that group) that is going to solve the issue. We used a similar idea, in our much simpler context, with the BERT baseline described in Section 5. However, in this paper we show how also with a limited number of examples, pre-trained LLMs can achieve similar performance.

Differently from these works, which are based on custom algorithms and models (which require a considerable effort for designing, implementing and testing), in our work we verify whether pre-trained LLMs can be used for this kind of task, even without fine-tuning. More generally, we exploit prompt engineering, which was designed precisely for obtaining the best results from these pre-trained models. Regarding this line of work, White et al. [23] provide a pattern catalog to solve common problems when conversing with a LLM. They propose 17 patterns which allow the users to better handle the input to give to the model, the output structure and format, possible errors in content, i.e. invented answers based on unverified facts, the prompt and how it can be improved to receive better responses and the interaction between the user and the model and the context needed by the model to generate a better response.

Moreover, Reynolds et al. [24] showed that zero shot learning (i.e. without any example provided to the model) with a good prompt can outperform a standard few shots approach (i.e. with some examples); to do so they introduced the concept of *meta-prompt*, that seeds the model to generate its own natural language prompt to solve the task. A similar result has been achieved by Zhou et al. [25], where the authors propose *Automatic Prompt Engineer*, a framework for automatic instruction generation and selection. In their method the authors optimized the prompt by searching over a pool of instruction candidates proposed by an LLM in order to maximize a chosen score function.

The LLMs, in particular ChatGPT, have been proven very effective in solving specific NLP tasks on general domains. Even in specific domains, such as public health [26, 27, 28], environmental problems [29] or legal rulings and laws [30, 31], the LLMs achieve acceptable performance. To the best of our knowledge, there are currently no applications of GPT-based models in the context of workplace security.

3. Available data

Since the release of DynDevice in 2011, Mega Italia Media started facing the problem of assisting and supporting end users. Originally, this service was provided by phone calls or emails but, with the strong spread of the platform over the years, these channels were soon saturated. To help the company operators to solve the users problem, in 2013 the company developed a ticketing system; this new feature allowed the company to keep track of all the tickets opened, memorizing the status of the user's request and who was in charge to solve the problem. Moreover, this system kept record of all the conversations between users and operators.

Overall, the company has received more than 10000 tickets in the last 10 years. However, in the last period several new features and services (such as multiple interface changes, a videocall system and several AI applications) were introduced, and therefore we decided to consider in our dataset only the tickets received in the last months, which are about 1300.

Furthermore, to build our dataset we decided only to keep the significant information: the ticket *category*, *object* and *description* and the *area* who solved the ticket; other information such as dates and identifiers has been removed. In the following we provide an example of a ticket; please note that this example has been translated, since all our tickets are written in Italian.

Example 1.

CATEGORY: *G. More on e-Learning/training solutions*

SUBJECT: *Certificate with exam in presence HTML5*

DESCRIPTION: *"Certificate with Examination in Attendance" I turned it into HTML 5. Everything is fine except for the column rows that do not appear. What could be the problem? Thank you*

AREA: *SW*

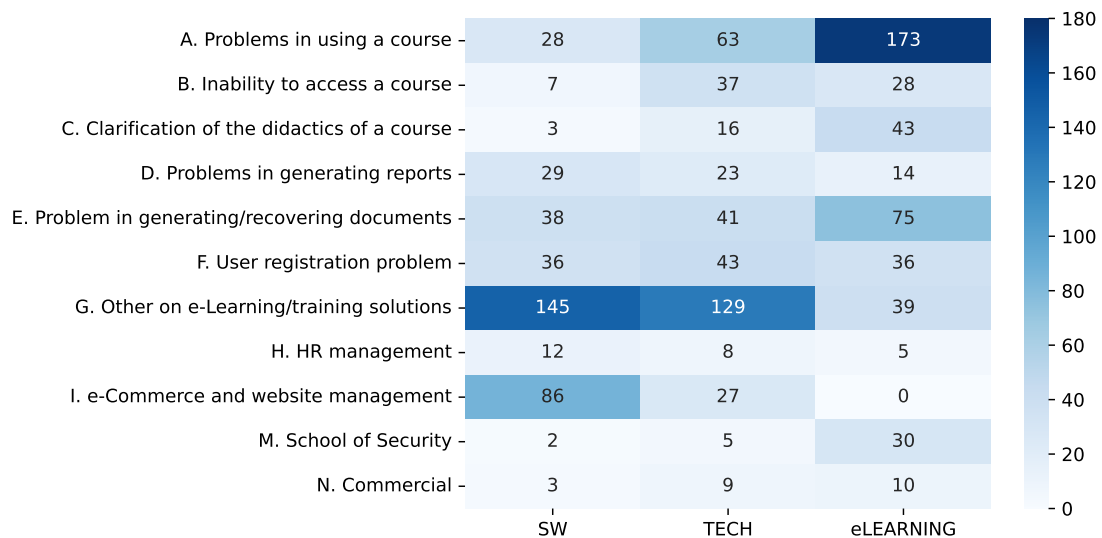


Figure 1: Cross table between ticket categories and macro areas who are in charge of solving the ticket.

The category field contains one of the 11 predetermined categories decided by the company. A complete categories list is reported in Figure 1. The object and the description are text fields that require a slight pre-process in order to remove the HTML tags, the URLs and other special characters that can harm the classification process. Each ticket is assigned to a specific operator who is in charge of solving the raised issue. However, all the operators can be aggregated in three main macro areas:

- **eLEARNING**: which handles the problems on the courses provided to the end users;
- **TECH**: which solves the technical issues about the platform;
- **SW**: that removes bugs and other software issues.

Each area manager assigns the ticket to a single operator suited to handle the case.

In the left part of the Figure 2, we reported the ticket distribution among the macro areas; as we can see the three classes are approximately equally distributed. On the contrary, as shown in the right part of Figure 2, the categories follow an unbalanced distribution, with category G being the most frequent, with more than 300 tickets compared to the least frequent. Another statistic that we extrapolate from the dataset is the cross tabulation between the categories and the area fields. As we can see in Figure 1, there is a strong correlation between some categories and some areas: the category A has a strong correlation with the eLEARNING area (and vice-versa), whereas both TECH and SW have a strong correlation with the category G. We expect tickets in these categories to be best assigned with a well constructed prompt containing this information. Other categories do not present any correlation, in some cases due to the low number of tickets, such as categories N and H. In other cases (such as categories F and E) the tickets are equally distributed between all the areas.

Finally, we decided to sample with stratification, following the area distribution, approximately 250 tickets to build 5 different test sets; this way, each test set contains 18 tickets for eLEARNING, 15 tickets for TECH e 14 tickets for SW. The tickets sampled with this strategy retain also the categories distribution and the cross correlation discussed above.

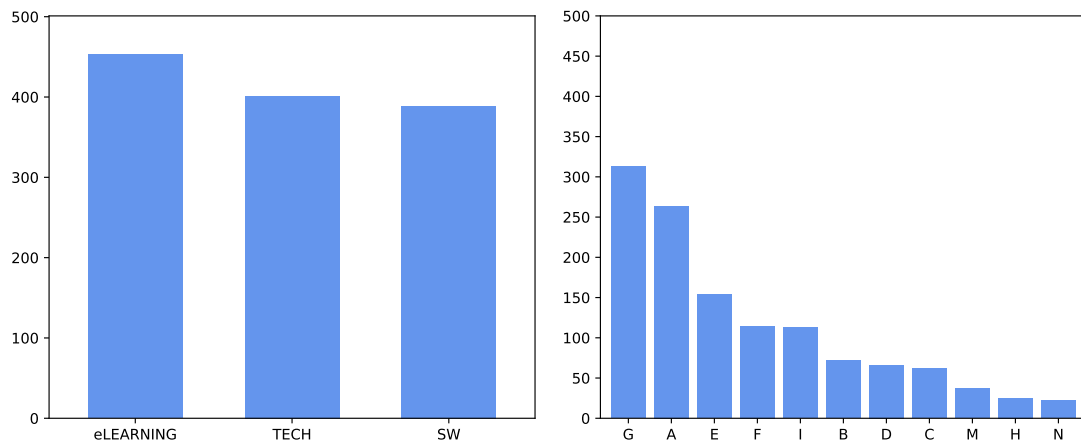


Figure 2: On the left, an histogram showing ticket distribution by the macro areas (i.e. our classification labels). On the right, an histogram showing the ticket distribution for each category considered.

4. Prompts and methods for automatic ticket assignment

The base task to solve for the automatic ticket assignment is a multi-class classification, into which we have to choose which group (eLEARNING, TECH or SW) will receive the ticket. To solve this task, we decided to exploit the Python version of the OpenAI Chat Completion API⁴, which allows interaction with pre-trained LLMs. For each call, the API requires several parameters. The most relevant ones for our work are: the *model* we want to query, which in our case can be GPT-4 or GPT-3.5-turbo, and the *temperature*, a decimal number between 0.0 and 2.0 (default 1.0), which controls randomness (higher values) or determinism (lower values) in the response generated. To have as much determinism as possible, in all trials we set temperature to 0.0. The corpus of the API request is composed by two messages: the *system prompt*, that contains the description of the task and the information to solve it, and the *user prompt*, i.e. the ticket to classify.

We tested GPT in three ways: zero shot learning, few shots learning, and ensemble learning. For the **zero shot learning** scenario, we provided to the model insightful information in the system prompt and no examples; all the information was extracted from the database described in Section 3. These are the zero shot prompts we implemented:

- **Baseline:** it describes the task, provides the basic information and imposes to GPT to answer only with the macro area name. From here on, each subsequent prompt is to be considered concatenated with this one.

Example: *You are the manager of a service center whose task is to divide tickets between the various human operators. The available operators, contained in square brackets are as follows: [eLEARNING, TECH, SW]. The tickets are divided into categories, listed by letters of the alphabet, contained in round brackets, which are as follows: (A. Problems in using a course, B. ...). Each ticket consists of a subject and a description. Your task is to assign the ticket to the most suitable operator, answering only with the operator's name.*

- **Human:** it contains insightful information, provided by a human employee, on the role of the three areas and the problems solved.

Example: *SW handles technical problems with software code, new customisation and developments and ICT (Information and Communication Technologies) and SEO (Search Engine Optimisation) issues.*

The same information is provided for the other 2 areas.

- **Categories:** it contains information related to the assignment of the tickets aggregated by category. For each category, we extract the percentage of tickets solved by each area belonging to that category (which can be seen in Figure 1). For instance, for the category A we wrote:

Example: *66% of the tickets in the category "A. Problems in using a course" are assigned to eLEARNING, 24% to TECH and 11% to SW.*

The same information is provided for the other 10 categories.

- **Areas:** it contains information about the assignment of tickets aggregated by areas. From the Figure 1, for each area, we extract the percentage of tickets belonging to each category

⁴<https://platform.openai.com/docs/api-reference/chat>

solved by the area. For instance, for the area eLEARNING we have:

Example: *To eLEARNING is assigned 38% of the tickets in category A, 8% in B, 9% in C, 3% in D, 17% in E, 8% in F, 8% in G, 1% in H, 0% in I, 7% in M and 2% in N.*

The same information is provided for the other 2 areas.

- **Summaries:** for this prompt we asked GPT-4 to summarize the issues raised by the users in 10 tickets for each area; these tickets were randomly sampled from the training dataset. For instance, for the area eLEARNING we have the following result:

Example: *eLEARNING solves problems related to the activation of courses, changes of certificates, platform access problems, cancellation of courses, issuing of certificates, downloading of certificates and approval of enrolment forms.*

The same summary is generated by GPT-4 for the other two areas and used by GPT in the system prompt. Please note that these summaries are generated separately by the model. At the moment of the ticket assignment, the LLM receives the summary without any knowledge of the 10 examples used for generating it.

As a second approach we implemented the **few shots learning**. The basic idea is to give to the model some examples of classification so it can understand the pattern, generalize it and then apply what it has learned to test instances. This approach has achieved outstanding results in a lot of NLP tasks [6, 8]. All our few shots approaches use the Baseline Prompt, without additional information. Instead, we provided to the model some examples in the same format expressed in the Example 1, with the correct macro area assigned.

The last approach, the **ensemble learning**, leverages the best results obtained in the past trials to try to use all the information at our disposal. Thus, for each ticket, we made n calls to the OpenAI API with different prompts, keeping model and temperature unchanged. Each result is counted as a vote, with no specific weight assigned, and the area with the most votes has the ticket assigned. In the event of a tie, the ticket is randomly assigned to one of the areas with the most votes. In our experimental evaluation we use an ensemble made by **three** and **four** prompts.

5. Experimental results

In this section, we report the results of our experiments. For each trial, in Table 1 we report the mean and the standard deviation over the five test sets for two metrics: the accuracy and the macro F1 score, both expressed between 0.0, the worst, and 1.0, the best.

In addition to the approaches described in the previous section, to provide a baseline for comparison with state-of-the-art approaches, we tested how a classical approach with BERT solves the task. Starting from a pre-trained Italian version of BERT available on HuggingFace⁵, we fine-tuned the model on this specific task on 907 samples for 20 epochs and we took the best performing model on a validation set made by 101 examples. To perform the classification, we add to BERT a simple feedforward layer with three neurons preceded by a 0.1 dropout layer. In training we used the Binary Cross Entropy loss function, the AdamW optimizer, with learning rate set to $2e-5$; we set decay to 0.01 and batch size to 32. We fed this classification model with

⁵dbmdz/bert-base-italian-uncased

Approach	Prompt	Accuracy	Macro F1
ZS	Baseline	0.34 ± 0.05	0.29 ± 0.05
ZS	Summaries	0.50 ± 0.07	0.48 ± 0.06
ZS	Categories	0.53 ± 0.06	0.49 ± 0.09
ZS	All Information	0.54 ± 0.05	0.49 ± 0.07
ZS	Areas	0.56 ± 0.03	0.50 ± 0.04
ZS	Human	0.57 ± 0.03	0.54 ± 0.04
FS	One Example	0.19 ± 0.05	0.14 ± 0.05
FS	Three Examples	0.44 ± 0.02	0.36 ± 0.04
FS	Five Examples	0.44 ± 0.05	0.34 ± 0.05
FS	Ten Examples	0.56 ± 0.04	0.51 ± 0.04
BERT	Only Ticket	0.62 ± 0.07	0.62 ± 0.07
BERT	Full	0.65 ± 0.07	0.65 ± 0.07
EL	Three Prompts	0.57 ± 0.05	0.52 ± 0.06
EL	Four Prompts	0.61 ± 0.05	0.55 ± 0.08

Table 1

Experimental results of our methods. In the Approach column, we specify if we use a zero-shot approach (ZS), a few-shot (FS) a fine-tuned BERT (BERT) or an ensemble learning approach (EL). In the Prompt column, we specify the additional configuration of the approach (as described in Section 4). For both Accuracy and Macro F1 we report their mean \pm their standard deviation calculated over five different test sets.

two types of input: one containing just the description of the ticket and one with all its three properties: category, object and description. The results obtained by the zero shot learning with no additional information (0.34 accuracy and 0.29 macro F1 score), and by BERT (0.65 accuracy and 0.65 macro F1 score), constitute the baselines of our framework.

As regarding the zero shot approach, when we start adding information to the base prompt, the accuracy improves. The two trials that leverage the correlation between the area and the category reach accuracy 0.53 for the Categories Prompt and 0.56 for the Areas Prompt. The reason for this behavior could be due to the length of the prompts: the first prompt is, more or less, 200 tokens long and some information is forgotten or ignored by the LLM. Also, in these two cases we provided only information about the categories and no information about the ticket description. This information is contained in the other two prompts: the Human and Summaries prompts. For the first case, the model reaches about 0.57 accuracy and an higher macro F1 score (0.54) and with the lowest standard deviation. For the second prompt, the accuracy drops to 0.50 with an higher standard deviation (0.07), probably due to the fact that the summaries are obtained with 10 tickets only, and the accuracy changes depending on whether the tickets in the dataset are more or less similar than those used for the summaries. When

we pass all the information to the model the performance is lower (0.54 accuracy), probably because the prompt reaches a critical length.

For the few shots approaches, the results are not particularly good. The single example case manages to perform even worse with respect to the baseline, with an accuracy of approximately 0.19. However, as it can be seen from the Table 1, increasing the number of examples improves the accuracy; with 3 and 5 examples, we reach 0.44 accuracy. This is probably due to the fact that the tickets are very varied and a small part of them does not represent the totality of issues addressed by a macro area. Only with 10 examples we get 0.56 (with a standard deviation of 0.04), approximately the same results obtained by the zero shot approaches.

The only approach that almost reaches the BERT baseline is the ensemble learning. In this case, using the best single information zero shot trials, combined with a voting system, helps the performances to get 0.57 with three prompts (Human, Areas and Categories) and 0.61 with all four prompts; in these cases the standard deviation is lower w.r.t. the BERT baseline. Also, in these scenarios, we can use all the information described in the zero shot approach without increasing the prompt length, unlike the full zero-shot case. Probably, with different tie breaking strategies (perhaps based on prediction probabilities) even better performances can be achieved; unfortunately, at the moment the Chat Competition OpenAI API does not provide any probabilistic information.

The second experiment focused on comparing the performance of the two main GPT models made available by OpenAI: GPT-4, and its predecessor GPT-3.5. Although this can be interesting from the researcher's perspective, this comparison has also an economic motivation. In fact,

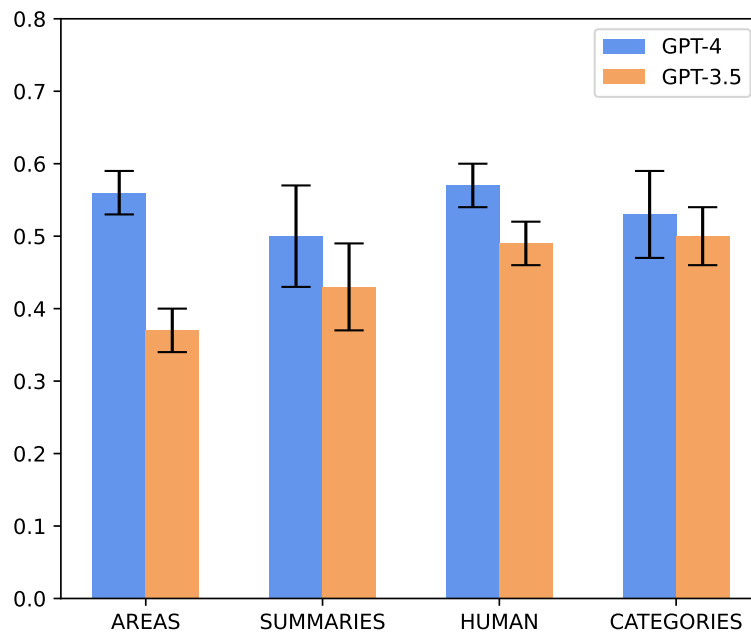


Figure 3: Accuracy comparison between GPT-3.5 and GPT-4 for the four zero shot prompts. The bars represent the mean accuracy and the black lines the standard deviation range.

according to the API pricing, invoking GPT-3.5 costs more than 20 times less than GPT-4. In this experiment, we keep the same prompts and the other hyperparameters, modifying just the model. As we can see in Figure 3, the difference in performance is noticeable. The best result obtained by GPT-3.5 is with the Categories Prompt, with an accuracy of 0.50, less than 5 points lower than GPT-4; the Human prompt loses approximately 10 points, while the Summaries prompt drops 15 points. The worst results is obtained by the Areas prompt, with an accuracy of 0.37 and a drop of 20 points. This behaviour is probably due to the long list of percentages for each single areas and which can confuse the model. No difference was found in the standard deviations. An average loss of 6 accuracy points between the 3 best performing prompts (thus excluding Areas) could still justify the cost of using GPT-4 over its older but cheaper counterpart.

6. Conclusions and Future work

In this work, we have shown an application of pre-trained Large Language Models for the automatic ticket assignment, based on the real-world data provided by Mega Italia Media, a company that provides IT solutions in the e-learning context.

In particular, we analyzed how different prompts, with more or less information and examples, influence the performance on this task. The experimental evaluation shows that in our context the classical few-shots approach does not provide a considerable improvement. This is probably because a limited number of examples can't capture the overall variety of tickets that the company receives. Instead, a zero-shot approach definitely improves the performance since it considers more information related to the overall context of the application (for instance a human description of the different classes or the percentage of instances belonging to each category). The best results are obtained by the ensemble methods involving three and four prompts. Their results almost reach the ones obtained by a BERT model specifically fine-tuned for this task. In contrast to BERT, which uses more than 1000 tickets between training and validation, this approach uses none. This approach can certainly be more efficient in scenarios where there is a little amount of data available.

As future work, we aim to study other prompting techniques, such as the chain-of-thought prompting [32] or prompts automatically generated [15]. Moreover, we would like to explore the performance of other LLMs, especially testing the open source ones, such as OpenAssistant, Dolly, GPT-J or GPT-NeoX. This could lead to a more detailed study, not only in terms of measuring the performance of each model, but also trying to understand what the models know in this field (which involves information technology, programming languages, e-learning, etc.) how this knowledge is stored in such models [33, 34], and focus on their explainability, analysing the behaviour of the attention mechanisms [35, 36] and prevent unwanted or discriminatory behaviour [37].

References

- [1] E. Bassignana, D. Brunato, M. Polignano, A. Ramponi, Preface to the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI), in: Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located with

- 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023), 2023.
- [2] A. Revina, K. Búza, V. G. Meister, IT ticket classification: The simpler, the better, *IEEE Access* 8 (2020) 193380–193395.
 - [3] K. Khowongprasoed, T. Titijaroonroj, Automatic thai ticket classification by using machine learning for IT infrastructure company, in: 19th International Joint Conference on Computer Science and Software Engineering, JCSSE 2022, Bangkok, Thailand, June 22-25, 2022, IEEE, 2022, pp. 1–6.
 - [4] M. Marcuzzo, A. Zangari, M. Schiavinato, L. Giudice, A. Gasparetto, A. Albarelli, A multi-level approach for hierarchical ticket classification, in: Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022), Association for Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 201–214.
 - [5] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.
 - [6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, *OpenAI blog* 1 (2019) 9.
 - [7] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, 2022.
 - [8] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, *ACM computing surveys (csur)* 53 (2020) 1–34.
 - [9] N. Arici, A. E. Gerevini, L. Putelli, I. Serina, L. Sigalini, A bert-based scoring system for workplace safety courses in italian, in: AIXIA 2022 - Advances in Artificial Intelligence - XXIst International Conference of the Italian Association for Artificial Intelligence, AIXIA 2022, Udine, Italy, November 28 - December 2, 2022, Proceedings, volume 13796 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 457–471.
 - [10] N. Arici, A. E. Gerevini, M. Olivato, L. Putelli, L. Sigalini, I. Serina, Real-world implementation and integration of an automatic scoring system for workplace safety courses in italian, *Future Internet* 15 (2023). doi:10.3390/fi15080268.
 - [11] M. Zubani, L. Sigalini, I. Serina, L. Putelli, A. E. Gerevini, M. Chiari, A performance comparison of different cloud-based natural language understanding services for an italian e-learning platform, *Future Internet* 14 (2022) 62.
 - [12] OpenAI, GPT-4 technical report, *CoRR abs/2303.08774* (2023). URL: <https://doi.org/10.48550/arXiv.2303.08774>. doi:10.48550/arXiv.2303.08774. arXiv:2303.08774.
 - [13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.
 - [14] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know, *Trans. Assoc. Comput. Linguistics* 8 (2020) 423–438.

- [15] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 4222–4235.
- [16] P. Zicari, G. Folino, M. Guarascio, L. Pontieri, Discovering accurate deep learning based predictive models for automatic customer support ticket classification, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1098–1101.
- [17] W. Zhou, W. Xue, R. Baral, Q. Wang, C. Zeng, T. Li, J. Xu, Z. Liu, L. Shwartz, G. Y. Grabarnik, STAR: A system for ticket analysis and resolution, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, ACM, 2017, pp. 2181–2190.
- [18] P. Molino, H. Zheng, Y. Wang, COTA: improving the speed and accuracy of customer support through ranking and deep networks, in: Y. Guo, F. Farooq (Eds.), Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, ACM, 2018, pp. 586–595.
- [19] A. DeLucia, E. Moore, Analyzing HPC support tickets: Experience and recommendations, CoRR abs/2010.04321 (2020). [arXiv:2010.04321](https://arxiv.org/abs/2010.04321).
- [20] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, volume 32, 2014, pp. 1188–1196.
- [21] J. Han, A. Sun, Deeprouting: A deep neural network approach for ticket routing in expert network, in: 2020 IEEE International Conference on Services Computing, SCC 2020, Beijing, China, November 7-11, 2020, IEEE, 2020, pp. 386–393.
- [22] L. Feng, J. Senapati, B. Liu, Tadaa: real time ticket assignment deep learning auto advisor for customer support, help desk, and issue ticketing systems, CoRR abs/2207.11187 (2022).
- [23] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with chatgpt, CoRR abs/2302.11382 (2023).
- [24] L. Reynolds, K. McDonell, Prompt programming for large language models: Beyond the few-shot paradigm, in: Y. Kitamura, A. Quigley, K. Isbister, T. Igarashi (Eds.), CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8-13, 2021, Extended Abstracts, ACM, 2021, pp. 314:1–314:7.
- [25] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large language models are human-level prompt engineers, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net, 2023.
- [26] S. S. Biswas, Role of chat GPT in public health, *Ann Biomed Eng* 51 (2023) 868–869.
- [27] T. Mehmood, A. Gerevini, A. Lavelli, I. Serina, Leveraging multi-task learning for biomedical named entity recognition, in: M. Alviano, G. Greco, F. Scarcello (Eds.), AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings, volume 11946 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 431–444. URL: https://doi.org/10.1007/978-3-030-35166-3_31. doi:10.1007/978-3-030-35166-3_31.
- [28] T. Mehmood, A. E. Gerevini, A. Lavelli, I. Serina, Combining multi-task learning with

transfer learning for biomedical named entity recognition, in: M. Cristani, C. Toro, C. Zanni-Merk, R. J. Howlett, L. C. Jain (Eds.), Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES-2020, Virtual Event, 16-18 September 2020, volume 176 of *Procedia Computer Science*, Elsevier, 2020, pp. 848–857. URL: <https://doi.org/10.1016/j.procs.2020.09.080>. doi:10.1016/j.procs.2020.09.080.

- [29] J.-J. Zhu, J. Jiang, M. Yang, Z. J. Ren, Chatgpt and environmental research, *Environmental Science & Technology* (2023). doi:10.1021/acs.est.3c01818, PMID: 36943179.
- [30] J. H. Choi, K. E. Hickman, A. Monahan, D. B. Schwarcz, Chatgpt goes to law school, *Journal of Legal Education* (2023).
- [31] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-BERT: the muppets straight out of law school, *CoRR abs/2010.02559* (2020).
- [32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: *NeurIPS, 2022*.
- [33] L. Serina, L. Putelli, A. E. Gerevini, I. Serina, Synonyms, antonyms and factual knowledge in BERT heads, *Future Internet* 15 (2023) 230.
- [34] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, F. Wei, Knowledge neurons in pretrained transformers, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Association for Computational Linguistics, 2022, pp. 8493–8502.
- [35] L. Putelli, A. E. Gerevini, A. Lavelli, I. Serina, The impact of self-interaction attention on the extraction of drug-drug interactions, in: *CLiC-it, volume 2481 of CEUR Workshop Proceedings*, CEUR-WS.org, 2019.
- [36] L. Putelli, A. E. Gerevini, A. Lavelli, T. Mehmood, I. Serina, On the behaviour of bert’s attention for the classification of medical reports, in: *XAL.it@AI*IA, volume 3277 of CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 16–30.
- [37] M. Dusi, N. Arici, A. E. Gerevini, L. Putelli, I. Serina, Graphical identification of gender bias in BERT with a weakly supervised approach, in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), *Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), Udine, November 30th, 2022*, volume 3287 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 164–176. URL: <https://ceur-ws.org/Vol-3287/paper16.pdf>.