Can language models automate data wrangling?

Gonzalo Jaimovitch-López[1], César Ferri[1], José Hernández-Orallo[1], Fernando Martínez-Plumed[1,2], and María José Ramírez-Quintana[1]

[1]VRAIN. Universitat Politècnica de València, Spain
[2]European Commission, Joint Research Centre
{gonjailo,cferri,jorallo,fmartinez,mramirez}@dsic.upv.es

Abstract. The automation of data science and other data manipulation processes depend on the integration and formatting of 'messy' data. Data wrangling is an umbrella term for these tedious and time-consuming tasks. Tasks such as transforming dates, units or names expressed in different formats have been challenging for machine learning because (1) users expect to solve them with short cues or few examples, and (2) the problems depend heavily on domain knowledge. Interestingly, large language models today (1) infer from very few examples or even a short clue in natural language, and (2) integrate vast amounts of domain knowledge. It is then an important research question to analyse whether language models are a promising approach for data wrangling, especially as their capabilities continue growing. In this paper we apply different language model variants of GPT to data wrangling problems, comparing their results to specialised data wrangling tools, also analysing the trends, variations and further possibilities and risks of language models in this task. Our major finding is that they appear as a powerful tool for a wide range of data wrangling tasks, but reliability may be an important issue to overcome.

Keywords: Data Science Automation · Data Wrangling · Language Models.

1 Introduction

Data wrangling refers to repetitive and time-consuming data preparation tasks, including transforming data presented in different formats into a standardised form for easy access, understanding and analysis. The (semi-)automation of these manual and non-systematic tasks can impact the costs of data preparation significantly. If language models (on their own or integrated within other systems) are able to solve a significant proportion of these problems in the next years, the transformative effect on society and the marketplace would be huge, given how widespread these formatting chores happen (from spreadsheet manipulation to data science projects) [12].

One key difficulty of some data wrangling problems such as standardising a field into a single format stems in the context of interaction [30]. For automation to be really useful, the tool should be able to infer the transformation pattern from very few examples, and complete the rest automatically. The second challenge for data wrangling, and especially for data transformation into a common format lies in the myriad of different transformations and formats we may find depending on the domain of the data. For instance, in a date field, the day can be the first, second or third number, and these numbers can be delimited by different symbols. An AI system based only on basic string transformations may never find the right solution given just one

example without domain constraints or background knowledge, as the transformations needed for dates are very different from those used for addresses or emails.

There seems to be great potential in language models [2] for data wrangling precisely because they compress huge amounts of human knowledge about many different domains, and have recently shown reasonably good performance in contextualising this knowledge for few-shot inference [23, 25, 5, 13]. It is then very important to determine whether language models could be used in the future for data wrangling tasks, and whether they get better as the number of parameters increase, a question subject to recent debate [1, 29]. The applicability for language models for the automation of other parts of data science may also be affected by the progress in data wrangling, especially as we move towards more domain-dependent and more open-ended tasks, as shown in the quadrants of figure 1 in [9].

In this paper we test experimentally whether language models can be used to solve typical problems in data wrangling, using prompts that will have input-output examples and a single input ending the prompt, for which the language model will have to provide the output as a continuation of the prompt (e.g. `Input: 'marshap@gmail.com' \nOutput: 'marshap'\n\nInput: 'alant@hotmail.com'\nOutput:`). Concretely, we compare the inference power of GPT-3 with other specialised tools on a benchmark of simple data wrangling problems. To our knowledge, this is the first paper analysing the potential of language models for data wrangling systematically, looking for the influence of the size of the model, and the number of examples.

2 Related work

One of the challenges for the automation of data wrangling tasks is how the solutions can be built or selected from a vast space of transformations when only a few examples are provided by a user [4]. For this reason many data wrangling tasks are approached by combining the available information in the examples with some domain knowledge ("any information the learner has about the unknown transformation before seeing the examples" [27]), in an attempt to reduce the hypothesis space. Inductive Programming [15] has been a common paradigm to learn transformations from very few examples by incorporating prior knowledge about the domain in a declarative way. As this approach suffers from intractability when background knowledge becomes large, the use of ad-hoc domain-specific languages (DSLs) (see [8, 32]) restrict the search space, and has led to the first commercial products such as Microsoft Excel with FlashFill [15]. Even with domain-specific languages, many constraints on the transformations are added to make things work, or very specific collections of built-in facilities or functions. For instance, Amazon SageMaker Data Wrangler* contains over 300 built-in data transformations or even tools like Trifacta Wrangler [20] allows the user to define her own transformations. Many systems combine some of these ideas or apply ad-hoc optimisations [16, 3, 11, 22, 14, 28, 27]. On the other hand, in [7, 6], general-purpose inductive programming systems can still be used by using different domain-specific background knowledge that are selected or ranked from contextual information or meta-features about the examples to be transformed.

Language models are conceptually simple systems: they estimate the probability $p(y|x)$ of a given sequence of characters or tokens y following another sequence x,

*https://aws.amazon.com/es/sagemaker/data-wrangler/

in the spirit of efficient coding [26]. Today, these models are usually based on large deep learning architectures such as transformers (attention-based architectures, [31]), but they still estimate this same probability. They are trained over massive natural language corpora and hence exploit the extrinsic patterns borrowed from humans. However, beyond making plausible continuations following the inputs (the so-called 'prompts'), or as part of this capability, recent systems such as BERT [10], GPT-2 [24], GPT-3 [5], and PanGu-α [34] can also be employed as 'few-shot learners', trying to exploit intrinsic patterns in the prompt. Few-shot inference happens when the models are able to extrapolate from previous examples in the 'prompt', without being retrained or fine-tuned. Extensive experimental research is showing remarkable extrapolations [18, 17, 33, 19] from small prompts. The state of the art of language models suggest they can be a promising tool for data wrangling precisely because they (1) capture a wide range of domain background knowledge, and contextualise it to the problem quite effectively, without the need of extra knowledge (e.g., we do not have to tell them that '23/12/2021' is a date), and (2) they not only infer from very few examples (e.g., pairs of date transformations "`Input: 23/12/2021, Output: 12-23-2021`"), but we can also add hints to the prompt to make few-shot learning more effective, or even zero-shot learning possible (e.g., "`The conversion of 23/12/2021 into US format is:`").

3 Methodology: goals and experimental design

Our experimental goals are: (1) determine to which extent a state-of-the-art language model can obtain good results on these data wrangling problems under the few-shot setting, (2) analyse the effect of the number of instances given in the few-shot setting, (3) explore the effect of the number of parameters of the language model to better understand the future potential, (4) study the variation of performance for different domains, and (5) compare the results with some other systems specifically designed for data wrangling.

For the experimental setting, we employ the Data Wrangling Dataset Repository[†], a benchmark for data transformation problems. This repository includes many of the data wrangling tasks used in the literature (see, e.g., [11]) as well as new manually gathered tasks [7]. Overall, the repository contains 117 different tasks divided into 7 different domains (*dates*, *emails*, *freetext*, *names*, *phones*, *times* and *units*). For every task we find 6 examples composed by an input string and an output string. The output string corresponds to a corrected or modified version of the input string. In the appendix, we provide further details about the tasks in each domain in Table 2 and some illustrative examples in Table 3.

We use four versions of GPT-3 (a language model built and trained by OpenAI) of increasing capabilities: Ada, Babbage, Curie and DaVinci which line up closely with 350M, 1.3B, 6.7B, and 175B parameters, respectively[‡]. First, we analysed several possible prompts based on the recommendations stated in the OpenAI API[§]. As a result, the final prompt used in this work follows an input-output style, where the

[†] http://dmip.webs.upv.es/datawrangling/

[‡] For the sake of replicability and reproducibility, all the code and results can be found in https://github.com/gonzalojaimovitch/lm-dw

[§] https://openai.com/blog/openai-api/

string "`Input:`" is used to indicate the start of the input, and the string "`Output:`" is used to indicate the start of the output. The line break `\n` separates the input from the output of an example, as well as the examples in the prompt. The instance will have one (one-shot) or more (few-shot) input-output pairs, randomly selected (without considering the possible order sensitivity of GPT-3 [21]), of the same problem and domain, and one single input ending the prompt. The model language will have to provide the output by continuing the prompt. These are two one-shot examples (from different domains):

```
Input: ‘290386’\nOutput: ‘29-03-86’\n\nInput: ‘250374’\nOutput:
Input: ‘08:50-09:30’\nOutput: ‘09:30’\n\nInput: ‘09:50-08:30’\nOutput:
```

4 Results and Discussion

The results obtained by the four different models (Ada, Babbage, Curie and DaVinci) in the four learning settings analysed (from 1-shot to 4-shot) are depicted in Figure 1 (complete details in Table 4, Figure 4, Table 5 and Table 6 in the supplementary material). In general, the results show that language models can be employed to learn simple transformations from few examples, and, as expected, the accuracy improves when we provide more instances. We also see that, in general, the most powerful engine is DaVinci.

Nevertheless, the performance is not uniform across the analysed domains. We observe that the domain *emails* is the one where the GPT-3 models obtain the highest performance, whereas *units* is the domain with the lowest performance.

With the intention of getting more insight into how the models fail, we perform a fine-grain analysis of the 'units' domain. Table 1 includes examples of some of these tasks to better understand the differences in performance showed in Figure 2. The problems in tasks *getUnits-i* and *getValue-i* (see Table 2 for details) can be translated as "extracting a part of the string", a transformation that the GPT3 models can solve. Hence, we see that GPT-3 presents good results in domains where tasks can be solved by simple string transformations. However, *getSystem-i* and *convert-i* are much more complex tasks. Thus, *getUnits-i* requires the identification of the unit acronym (e.g., 'cl' for centilitres) and relating it with its unit system (e.g., volume), while *convert-i* needs to perform an arithmetic operation (e.g., a division), in addition to the identification of the conversion coefficient to the target unit (e.g., a coefficient of 1000 to convert milligrams into grams).

Table 1. Examples of problems in the domain 'units'.

Problem	**Input → Output**
'getUnits-1'	56.77cl → cl
'getValue-1'	56.77cl → 56.77
'getSystem-1'	56.77cl → Volume
'convert-1'	1441.8mg; g → 1.4418001

Finally, in order to compare the performance of GPT-3 with other data wrangling systems, we employ the subset of 26 problems for which there are results in the literature and a 1-shot setting, which is the same setting used by the other systems. We compare GPT-3 DaVinci and other data wrangling tools: FlashFill [15], TrifactaWrangler [22] and DBK [7]. The results (displayed in Figure 3) show that general-purpose

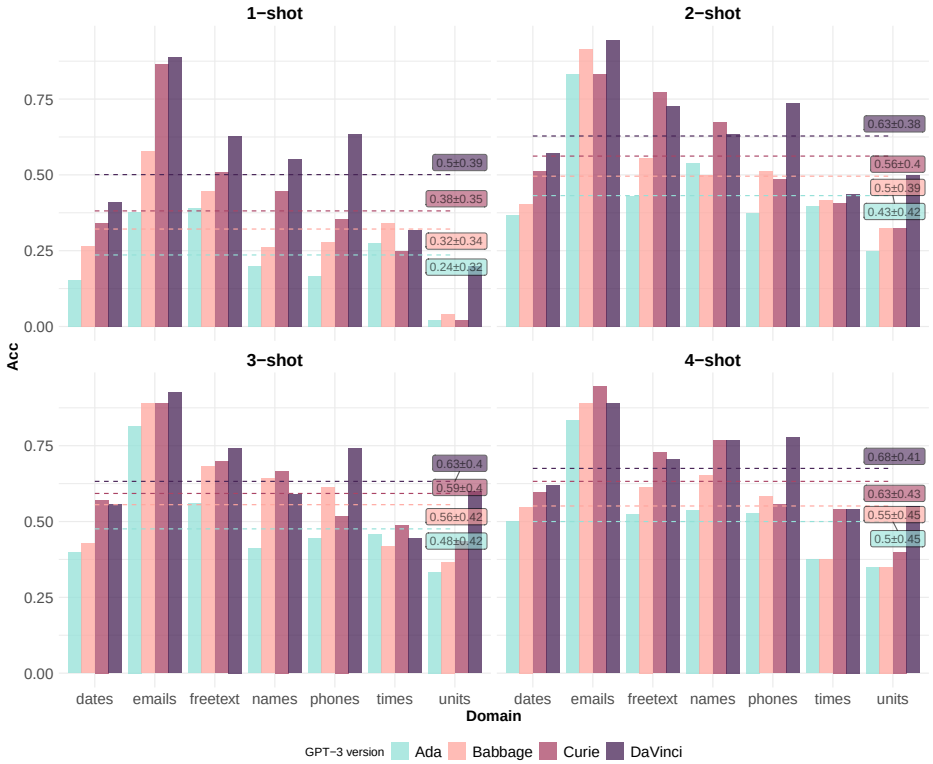

Fig. 1. Average results for the seven domains and the four versions of GPT-3. Each plot represents how many examples are given (from 1-shot to 4-shot). The dashed horizontal lines show the average results per system. Disaggregated results for all tasks shown in Table 4 and Figure 4 in the supplementary material.

language models are competitive with first-generation data wrangling tools such as FlashFill, and are getting closer in performance to more sophisticated tools such as DBK. Again, we see that the performance of the compared systems is related to the types involved in the target functions. The best results are obtained in domains where the problems are solved by simple string operations, while in other domains like *units* where some functions incorporate arithmetic the results are much worse. The exception is DBK that can induce the domain of the problem and then select proper base functions to address it.

5 Conclusions

Language models have recently disrupted artificial intelligence thanks to an unexpected abstraction capacity that has expanded their applicability to fields and problems not originally anticipated in their design. In this work, we have analysed different configurations and prompts, as well as the effect of the number of examples provided to see their performance for data wrangling problems. To our knowledge, this paper is the first one that explores the possibilities of language models for data wrangling problems. The results show the capacity of these systems to learn transformation functions from few examples. The performance of the studied language models is comparable

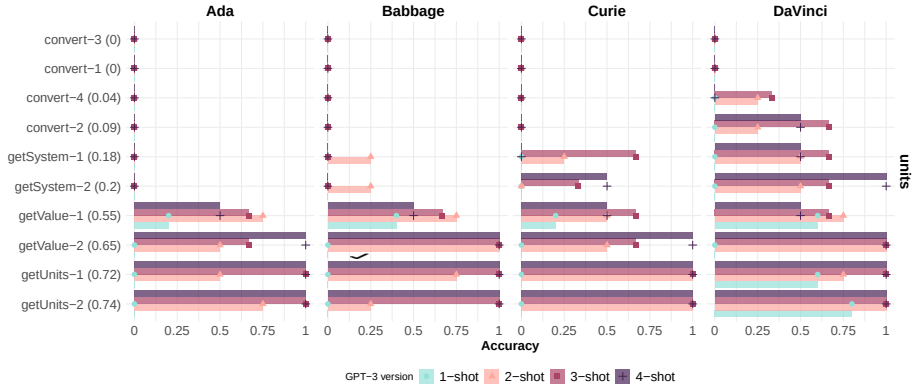

Fig. 2. Average accuracies for the tasks in the *units* domain for all GPT-3 systems and learning settings. Complete details for all domains and descriptions for all tasks in Figure 4 and Table 2, respectively, in the supplementary material.

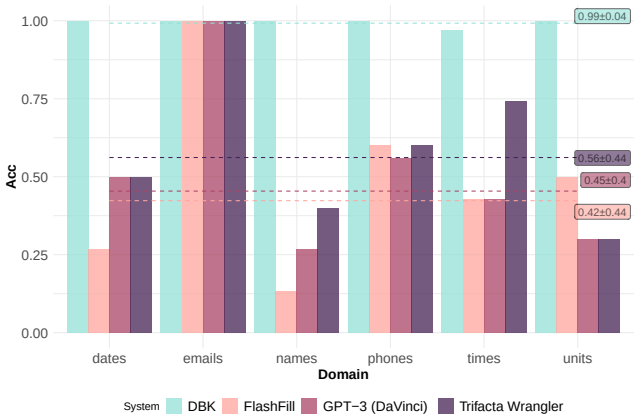

Fig. 3. Average results by GPT-3 (DaVinci version) compared to FlashFill [15], Trifacta Wrangler and DBK [7] for a 1-shot learning setting. Results of the compared systems are obtained from [7, 6]. The tasks addressed are a subset of those in Table 4. Coloured, horizontal lines show the average results per system across domains.

to well-known systems specialised in data wrangling. These results open a promising research direction to explore the possible applications of language models as APIs and specialised tools for data wrangling. This is not limited to data wrangling, but could well be used for other tasks in data science, especially those that can be learnt from very few examples and require extensive domain knowledge.

As future work, we are interested in exploring other scenarios where, additionally to the examples, we also give some textual hints about the problem. This is recommended in the OpenAI API documentation. We also plan to explore alternative prompts that could increase the learning capacity of language models. Finally, we also have detected some problems related to the reliability of the learned hypotheses by the language models. For instance, consistent hypotheses learned with few examples are later ignored when we provide more examples to the models.

Acknowledgements We thank Lidia Contreras for her help with the Data Wrangling Dataset Repository. We thank the anonymous reviewers for their comments. This work was funded by the Future of Life Institute, FLI, under grant RFP2-152, the MIT-Spain - INDITEX Sustainability Seed Fund under project COST-OMIZE, the EU (FEDER) and Spanish MINECO under RTI2018-094403-B-C32, Generalitat Valenciana under PROMETEO/2019/098 and INNEST/2021/317, EU's Horizon 2020 research and innovation programme under grant agreement No. 952215 (TAILOR) and US DARPA HR00112120007 ReCOG-AI. FMP acknowledges funding from the HUMAINT project by DG JRC of the European Commission.

References

1. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. p. 610–623. FAccT '21 (2021)
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. The journal of machine learning research **3**, 1137–1155 (2003)
3. Bhupatiraju, S., Singh, R., Mohamed, A.r., Kohli, P.: Deep api programmer: Learning to program with apis. arXiv preprint arXiv:1704.04327 (2017)
4. Bogatu, A., Paton, N.W., Fernandes, A.A.: Towards automatic data format transformations: Data wrangling at scale. In: British International Conference on Databases. pp. 36–48. Springer (2017)
5. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
6. Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J.: Automating common data science matrix transformations. In: ECMLPKDD workshop on Automating Data Science. ECML-PKDD '19 (2019)
7. Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., Ramírez-Quintana, M.J., Katayama, S.: Automated data transformation with inductive programming and dynamic background knowledge. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2019. ECML-PKDD '19 (2019)
8. Cropper, A., Tamaddoni, A., Muggleton, S.H.: Meta-interpretive learning of data transformation programs. In: Inductive Logic Programming. pp. 46–59 (2015)
9. De Bie, T., De Raedt, L., Hernández-Orallo, J., Hoos, H.H., Smyth, P., Williams, C.K.: Automating data science: Prospects and challenges. Communications of the ACM, to appear, arXiv preprint arXiv:2105.05699 (2021)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Ellis, K., Gulwani, S.: Learning to learn programs from examples: Going beyond program structure. In: IJCAI. pp. 1638–1645 (2017)
12. Furche, T., Gottlob, G., Libkin, L., Orsi, G., Paton, N.W.: Data wrangling for big data: Challenges and opportunities. In: EDBT. vol. 16, pp. 473–478 (2016)
13. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723 (2020)
14. Gulwani, S.: Automating string processing in spreadsheets using input-output examples. In: Procs. 38th Principles of Programming Languages. pp. 317–330 (2011)
15. Gulwani, S., Hernández-Orallo, J., Kitzelmann, E., Muggleton, S.H., Schmid, U., Zorn, B.: Inductive programming meets the real world. Communications of the ACM **58**(11), 90–99 (2015)

16. Ham, K.: Openrefine (version 2.5). http://openrefine. org. free, open-source tool for cleaning and transforming data. Journal of the Medical Library Association: JMLA **101**(3), 233 (2013)
17. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. ICLR (2021)
18. Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J.: Measuring mathematical problem solving with the MATH dataset. CoRR **abs/2103.03874** (2021), https://arxiv.org/abs/2103.03874
19. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282 (2020)
20. Kandel, S., Paepcke, A., Hellerstein, J., Heer, J.: Wrangler: Interactive visual specification of data transformation scripts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3363–3372. ACM (2011)
21. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786 (2021)
22. Petrova-Antonova, D., Tancheva, R.: Data cleaning: A case study with openrefine and trifacta wrangler. In: International Conference on the Quality of Information and Communications Technology. pp. 32–40. Springer (2020)
23. Puri, R., Catanzaro, B.: Zero-shot text classification with generative language models. arXiv preprint arXiv:1912.10165 (2019)
24. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
25. Schick, T., Schütze, H.: Exploiting cloze questions for few-shot text classification and natural language inference. arXiv preprint arXiv:2001.07676 (2020)
26. Shannon, C.E.: Communication theory of secrecy systems. The Bell system technical journal **28**(4), 656–715 (1949)
27. Singh, R., Gulwani, S.: Predicting a correct program in programming by example. In: International Conference on Computer Aided Verification. pp. 398–414. Springer (2015)
28. Singh, R., Gulwani, S.: Transforming spreadsheet data types using examples. In: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 343–356 (2016)
29. Tamkin, A., Brundage, M., Clark, J., Ganguli, D.: Understanding the capabilities, limitations, and societal impact of large language models. arXiv preprint arXiv:2102.02503 (2021)
30. Terrizzano, I.G., Schwarz, P.M., Roth, M., Colino, J.E.: Data wrangling: The challenging yourney from the wild to the lake. In: CIDR (2015)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
32. Wu, B., Szekely, P., Knoblock, C.A.: Learning data transformation rules through examples: Preliminary results. In: Information Integration on the Web. p. 8 (2012)
33. Xu, S., Semnani, S.J., Campagna, G., Lam, M.S.: AutoQA: From databases to QA semantic parsers with only synthetic training data. EMNLP (2020)
34. Zeng, W., Ren, X., Su, T., Wang, H., Liao, Y., Wang, Z., Jiang, X., Yang, Z., Wang, K., Zhang, X., et al.: Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. arXiv preprint arXiv:2104.12369 (2021)

Supplementary Material

Table 2. Datasets included in the data wrangling repository.

Task Description	Expected Output
Add punctuation	The date in numeric format split by a punctuation sign
Change format	The date in one particular format
Change Punctuation	The date in one particular format
Get Day	The day in numeric format
Get Day Ordinal	The day in numeric ordinal format
Get Month Name	The name of the month
Get Week Day	The name of the weekday
Reduce Month Name	The name of the month reduced to three letters
Set Format	The date split in DMY format
Generate Email	An email account created with the name and the domain
Get After At	Everything after the at symbol
Get Domain	The domain before the dot
Before At	Everything before the at symbol
After Symbol	Everything after a symbol
Between Symbols	Everything between a pair of symbols
Delete Punctuation	Remove punctuation
Delete Spaces	Remove blanks
Digit to End	Everything after the first digit if exists
First Character	Get first character
Get After Comma	Everything after a comma
Get Caps	Capitalise each word in a text
To Upper	Convert text to upper case
Add Title	The name with a title
Get Title	The title attached to the name, if exists
Generate Login	A login generated using the name
Reduce name	The name reduced before the surname(s)
Add Prefix by Country	Phone numbers with the prefix of the countries
Delete Parentheses	The list of phone numbers without parentheses
Get Number	A phone number presented in the string, if exists
Set Prefix	The list of phone numbers with the prefix
Set Punctuation	A phone number split by a punctuation sign
Add Time	The time increasing the hour by the integer
Append o'clock Time	The time appending an o'clock time
Append Time	The time appending the integer as new component
Convert Time	The time formatted to 24 hours format
Convert Time	The time formatted to a given format
Convert Time	The time formatted to 12 hours format
Convert Time	The time changed from the first time zone to the second
Delete Time	The time deleting the last component
Get Hour	The hour component
Get Minutes	The minutes component
Get Time	A time presented in the string
Convert Units	The value transformed to a different magnitude
Get System	The system represented by the magnitude
Get Units	The units of the system
Get Value	The numeric value without any magnitude

Table 3. Examples of data wrangling tasks of different domains included in the repository used for the experimentation.

Domain	Tasks	Example (*input* → *output*)
Dates	21	*74-03-31* → *31*
Email	10	*Jan.Kotas@litwareinc.com* → *litwareinc.com*
Freetext	25	*Association of Computational Linguistics* → *ACL*
Names	15	*Prof. Kathleen S. Fisher* → *Fisher, K.*
Phones	18	*John DOE 3 . . . [TS]865-000-0000 . . .* → *865-000-0000*
Times	24	*3:40 PM* → 15:40
Units	10	*12.20 dg* → *1220.0 mg*

Table 4. Results obtained by the different versions of GPT-3 models desegregated by learning strategy (from 1-shot to 4-shot learning), system, domain and data wrangling task.

		1-shot				2-shot				3-shot				4-shot			
Domain	**Problem**	**Ada**	**Babbage**	**Curie**	**DaVinci**	**Ada**	**Babbage**	**Curie**	**DaVinci**	**Ada**	**Babbage**	**Curie**	**DaVinci**	**Ada**	**Babbage**	**Curie**	**DaVinci**
	addPunctuation-1	0.00	0.60	0.60	0.60	0.00	0.75	1.00	1.00	0.33	0.67	1.00	1.00	0.50	1.00	1.00	1.00
	addPunctuation-2	0.00	0.40	0.40	0.40	0.50	0.50	0.50	0.50	0.67	0.33	0.67	0.67	0.50	0.50	0.50	1.00
	changeFormat-1	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.75	0.33	0.00	0.67	0.67	1.00	0.50	0.50	0.50
	changeFormat-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
	changeFormat-3	0.00	0.00	0.40	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	changeFormat-4	0.00	0.40	0.20	0.20	0.25	0.75	0.50	0.25	0.00	0.00	0.33	0.33	0.00	0.50	1.00	0.50
	changePunctuation-1	0.60	0.60	1.00	1.00	1.00	1.00	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	changePunctuation-2	0.80	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getDay-1	0.20	0.00	0.40	0.20	0.50	0.50	0.25	0.25	0.00	0.67	0.33	0.00	0.00	0.50	0.50	0.00
	getDay-2	0.00	0.00	0.00	0.00	0.75	0.25	0.25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getDay-3	0.20	0.00	0.60	1.00	0.75	0.75	0.75	1.00	0.67	0.67	1.00	1.00	1.00	1.00	1.00	1.00
	getDayOrdinal-1	0.20	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dates	getDayOrdinal-2	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.50	0.50
	getMonthName-1	0.00	0.40	0.40	0.80	0.25	0.50	0.50	0.75	0.00	0.67	0.67	0.67	0.50	0.50	0.50	0.50
	getMonthName-2	0.80	0.80	1.00	0.60	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getWeekDay-1	0.00	0.60	0.40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getWeekDay-2	0.20	0.80	0.40	1.00	0.25	0.00	1.00	1.00	0.33	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	reduceMonthName-1	0.20	0.20	0.40	0.20	0.25	0.25	0.75	0.75	0.67	0.33	0.67	1.00	0.50	0.50	0.50	1.00
	reduceMonthName-2	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.25	0.33	0.67	0.33	0.33	0.50	0.50	0.50	0.50
	setFormat-1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	setFormat-2	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	generate-1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	generate-2	0.00	0.60	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	generate-3	0.80	0.80	1.00	0.60	0.75	1.00	0.50	1.00	0.67	1.00	0.67	0.67	1.00	1.00	1.00	0.50
	getAfterAt-2	0.20	0.00	0.80	0.80	0.25	0.75	1.00	1.00	0.33	0.67	1.00	0.67	0.50	1.00	1.00	0.50
	getAfterAt-3	0.00	0.20	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
emails	getAfterAt-4	0.80	0.60	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getDomain-1	0.00	0.40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getDomain-2	0.20	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	someBeforeAt-NA	0.40	0.60	0.40	0.80	0.50	0.50	0.00	0.50	0.33	0.33	0.33	1.00	0.00	0.00	0.50	1.00
	afterSymbol-1	1.00	1.00	0.80	0.80	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	afterSymbol-2	0.60	1.00	0.80	0.40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	betweenSymbols-1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	betweenSymbols-2	0.00	0.80	0.40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	brackets-1	1.00	0.80	1.00	0.80	1.00	0.75	1.00	0.75	1.00	1.00	1.00	0.67	1.00	1.00	1.00	1.00
	brackets-2	0.60	0.40	0.80	0.60	1.00	0.25	0.75	0.25	1.00	1.00	1.00	0.67	1.00	1.00	1.00	1.00
	deletePunctuation-1	0.00	0.00	0.00	0.60	0.00	0.25	0.25	0.50	0.00	0.33	0.33	0.33	0.00	0.00	0.00	0.00
	deletePunctuation-2	0.40	0.40	0.40	0.40	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
	deletePunctuation-3	0.20	0.40	0.00	0.00	0.00	0.25	1.00	0.50	0.00	0.33	0.67	0.33	0.00	0.50	1.00	0.00
	deletePunctuation-4	0.00	0.00	0.40	0.60	0.00	0.00	0.50	0.50	0.33	0.33	0.33	0.33	0.00	0.00	0.00	0.00
	deleteSpaces-2	0.00	0.00	0.40	0.40	0.00	0.00	0.25	0.50	0.00	0.00	0.33	0.67	0.00	0.00	0.00	0.00
	digitToEnd-1	0.80	0.80	0.80	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
freetext	firstCharacter-1	0.00	0.40	0.40	0.60	0.25	0.25	0.75	0.75	0.67	0.67	0.00	0.67	0.00	0.00	0.50	1.00
	firstCharacter-2	0.00	0.00	0.00	0.20	0.00	0.50	0.75	0.75	0.00	0.67	0.67	0.67	0.50	0.50	0.50	0.50
	getAfterComma-1	1.00	0.80	0.20	1.00	0.75	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getAfterComma-2	1.00	1.00	0.60	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getBetweenCommas-1	1.00	1.00	1.00	0.80	1.00	1.00	0.75	0.75	1.00	1.00	0.67	1.00	1.00	1.00	1.00	1.00
	getCaps-1	0.00	0.00	0.00	0.80	0.00	0.25	1.00	0.75	0.67	1.00	1.00	0.67	0.50	1.00	1.00	1.00
	getCaps-2	0.00	0.00	0.40	0.80	0.00	0.50	0.50	1.00	0.33	0.67	0.33	1.00	0.50	0.50	0.00	0.50
	getCaps-3	0.00	0.00	1.00	0.80	0.00	0.75	1.00	1.00	0.33	1.00	1.00	1.00	0.00	1.00	1.00	1.00
	toUpper-1	0.00	0.00	0.40	0.20	0.00	0.50	0.50	0.50	0.00	0.00	0.33	0.33	0.00	0.00	0.50	0.50
	toUpper-2	0.00	0.00	0.40	0.00	0.00	0.00	0.75	0.25	0.00	0.00	0.67	1.00	0.00	0.00	0.50	1.00
	addTitle-1	0.60	0.40	0.40	0.60	0.75	0.75	0.75	0.75	0.67	0.67	0.67	0.33	1.00	1.00	1.00	1.00
	addTitle-2	0.40	0.40	0.40	0.40	0.50	0.25	0.50	0.00	0.33	0.33	0.33	0.33	0.50	0.00	0.50	0.50
	getTitle-1	0.00	0.00	0.20	0.20	0.00	0.00	0.50	0.75	0.00	0.33	0.67	0.67	0.00	0.50	1.00	1.00
	getTitle-2	0.20	0.00	0.40	1.00	0.75	0.75	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	login-1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	login-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	reduceName-1	0.40	0.80	0.60	0.80	1.00	0.50	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
names	reduceName-2	0.60	0.80	1.00	1.00	1.00	0.75	1.00	1.00	1.00	0.67	1.00	1.00	1.00	0.50	1.00	1.00
	reduceName-3	0.20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	1.00	1.00	1.00	0.50	1.00	1.00	1.00
	reduceName-5	0.00	0.00	0.60	0.80	0.75	0.75	0.75	0.50	0.67	1.00	0.67	0.67	1.00	1.00	1.00	0.50
	reduceName-6	0.00	0.00	0.20	0.60	0.75	0.50	0.50	0.50	0.67	0.67	0.33	0.33	1.00	0.50	0.50	1.00
	reduceName-8	0.00	0.00	0.40	0.20	0.00	0.25	0.75	1.00	0.00	0.67	1.00	0.33	0.00	1.00	1.00	1.00
	reduceName-9	0.20	0.00	0.60	0.60	0.50	1.00	1.00	1.00	0.33	1.00	1.00	1.00	0.00	1.00	1.00	1.00
	countryPrefix-1	0.00	0.20	0.00	1.00	0.00	0.75	0.25	1.00	0.00	0.67	0.33	1.00	0.00	1.00	1.00	1.00
	countryPrefix-2	0.00	0.00	0.00	0.60	0.00	0.00	0.25	1.00	0.00	0.33	0.00	1.00	0.00	0.00	0.00	1.00
	countryPrefix-3	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.50
	countryPrefix-7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	countryPrefix-8	0.40	0.80	0.40	1.00	0.00	0.75	0.75	1.00	0.33	0.67	0.67	1.00	1.00	1.00	1.00	1.00
	countryPrefix-9	0.00	0.80	0.60	1.00	0.00	0.75	0.50	1.00	0.33	1.00	0.67	1.00	0.50	1.00	1.00	1.00
	deleteParentheses-1	0.00	0.00	0.40	1.00	1.00	0.75	0.75	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	1.00
	deleteParentheses-2	0.20	0.40	0.20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getNumber-1	0.40	0.60	0.60	0.80	0.50	0.50	0.25	0.75	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getNumber-2	0.40	0.60	0.20	0.20	0.25	0.25	0.00	0.25	0.33	1.00	0.00	0.00	0.50	0.50	0.00	0.50
phones	setPrefix-1	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	setPrefix-2	0.00	0.20	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	setPrefix-3	1.00	0.60	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	setPrefix-4	0.40	0.20	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	setPrefix-5	0.20	0.20	0.20	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	setPrefix-6	0.00	0.00	0.20	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	setPunctuation-1	0.00	0.20	0.20	0.60	0.00	0.25	0.25	0.75	0.00	0.00	0.33	1.00	0.00	0.00	0.00	1.00
	setPunctuation-2	0.00	0.20	0.40	0.80	0.00	0.25	0.75	1.00	0.33	0.33	0.33	1.00	0.50	0.50	0.50	1.00
	addTime-1	0.20	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	addTime-2	0.00	0.00	0.00	0.80	0.25	0.25	0.25	0.50	0.33	0.33	0.33	0.67	0.00	0.00	0.00	0.50
	appendTime-1	0.40	0.40	0.40	0.20	0.75	0.25	0.25	0.25	0.67	0.67	0.67	0.67	0.50	0.50	0.50	0.50
	appendTime-2	0.80	0.80	0.80	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	appendTime-3	0.80	0.40	0.80	0.80	0.75	0.75	0.75	0.75	0.67	0.67	0.67	0.67	0.50	0.50	0.50	0.50
	appendTime-4	0.80	0.80	0.80	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	convert-1	0.40	0.20	0.20	0.20	0.25	0.25	0.50	0.50	0.67	0.33	0.33	0.33	0.50	0.50	0.50	0.50
	convert-10	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	convert-2	0.20	0.40	0.00	0.00	0.25	0.25	0.25	0.50	1.00	0.67	1.00	0.67	1.00	1.00	1.00	1.00
	convert-3	0.20	0.20	0.20	0.20	0.25	0.25	0.25	0.50	0.33	0.33	0.33	0.33	0.00	0.00	0.00	0.00
	convert-4	0.20	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.33	0.00	0.50	0.00	0.50	0.00
	convert-5	0.20	0.20	0.20	0.20	0.50	0.25	0.50	0.50	0.33	0.33	0.33	0.33	0.00	0.00	0.50	0.00
	convert-6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50
times	convert-7	0.20	0.20	0.00	0.00	0.50	0.50	0.00	0.00	0.33	0.33	0.33	0.33	0.00	0.00	0.00	0.00
	convert-8	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	convert-9	0.20	0.20	0.00	0.00	0.50	0.50	0.25	0.00	0.33	0.33	0.33	0.33	0.00	0.00	0.00	0.00
	deleteTime-1	0.00	0.00	0.20	0.00	0.25	0.25	0.50	0.50	0.33	0.33	0.33	0.67	0.50	0.50	0.50	0.50
	deleteTime-2	0.20	0.20	0.20	0.20	0.00	0.00	0.00	0.00	0.33	0.00	1.00	0.00	0.00	0.00	1.00	1.00
	getHour-1	0.20	0.20	0.20	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
	getHour-2	0.00	0.60	0.20	0.20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getMinutes-1	0.20	0.80	0.20	0.60	0.50	0.75	0.75	0.75	0.33	0.67	0.67	0.67	0.50	1.00	1.00	1.00
	getMinutes-2	0.20	0.20	0.20	0.20	0.00	0.75	0.50	0.75	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00
	getTime-1	0.20	0.60	0.60	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getTime-2	1.00	1.00	0.80	1.00	0.75	1.00	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	convert-1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	convert-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.50
	convert-3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	convert-4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00
	getSystem-1	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.50	0.00	0.00	0.67	0.67	0.00	0.00	0.00	0.50
units	getSystem-2	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.50	0.00	0.00	0.33	0.67	0.00	0.00	0.50	1.00
	getUnits-1	0.00	0.00	0.00	0.60	0.50	0.75	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getUnits-2	0.00	0.00	0.00	0.80	0.75	0.25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	getValue-1	0.20	0.40	0.20	0.60	0.75	0.75	0.50	0.75	0.67	0.67	0.67	0.67	0.50	0.50	0.50	0.50
	getValue-2	0.00	0.00	0.00	0.00	0.50	1.00	0.50	1.00	0.67	1.00	0.67	1.00	1.00	1.00	1.00	1.00

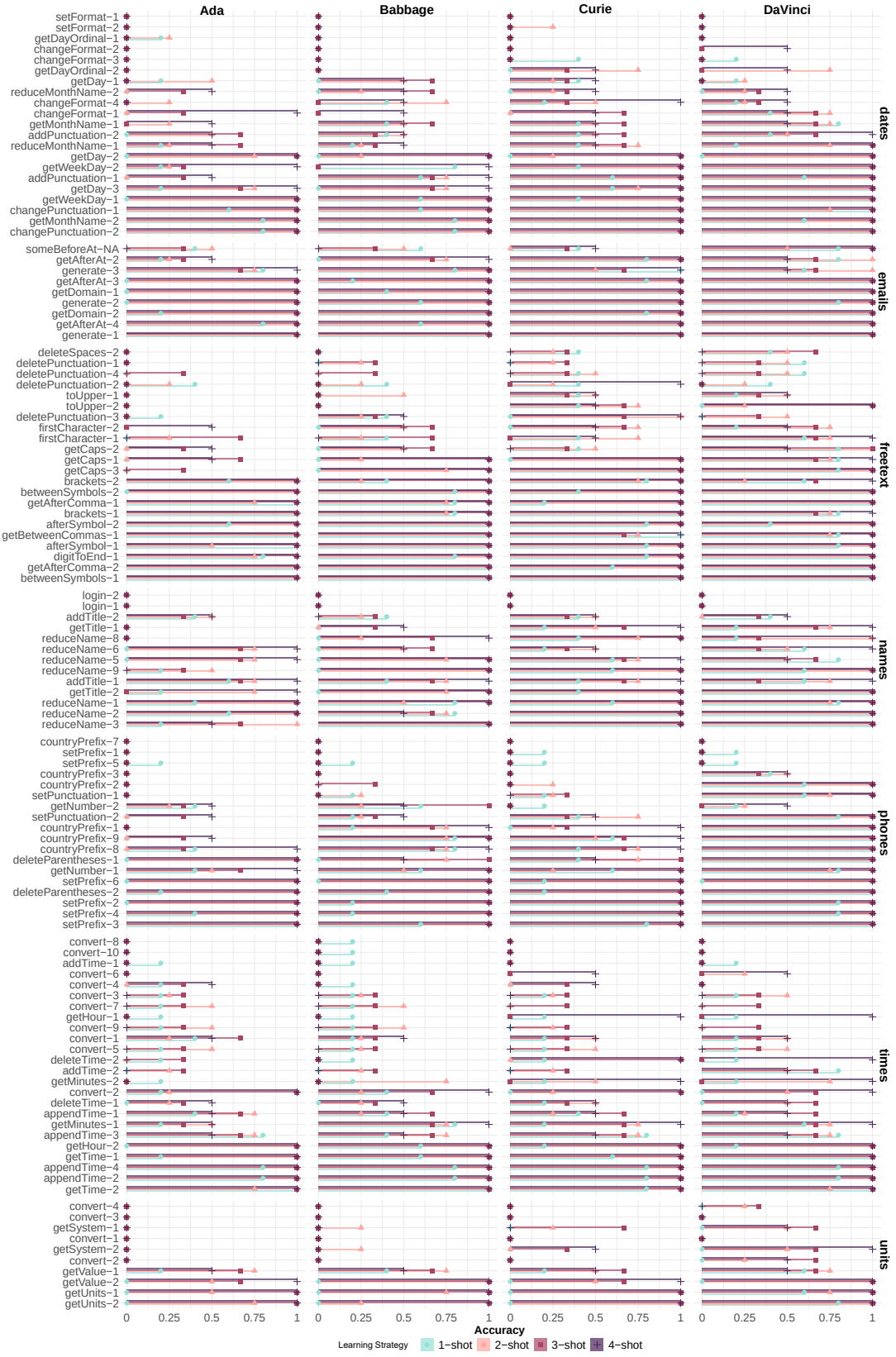

Fig. 4. Results obtained by the different versions of GPT-3 models disaggregated by learning strategy (from 1-shot to 4-shot learning), system, domain and data wrangling task. Tasks per domain sorted by average performance.

Table 5. Wrong answers provided by the GPT-3 DaVinci system in the 4-shot setting (examples based on large sentences are represented in Table 6).

Input 1	Input 2	Input 3	Output	Correct output	Problem	Domain
1990.18.01	-	-	01.18.1990	01/18/1990	changeFormat-1	dates
28-10-2001	-	-	10/28/2001	08/23/2010	changeFormat-2	dates
99/09/03	-	-	09/99/03	09/03/99	changeFormat-3	dates
06-04-2001	-	-	04/06/2001	06/04/2001	changeFormat-3	dates
08-07-2017	-	-	07-08-2017	08-07-2017	changeFormat-4	dates
31/05/17	-	-	17	31	getDay-1	dates
25-08-85	-	-	08	25	getDay-1	dates
31/03/75	-	-	31st	31th	getDayOrdinal-1	dates
09/11/53	-	-	11th	09th	getDayOrdinal-1	dates
05/09/2008	-	-	5th	05th	getDayOrdinal-2	dates
Thursday, October 31, 1985	-	-	October	Thursday	getMonthName-1	dates
11/06/2016	-	-	Jun	Nov	reduceMonthName-2	dates
8072017	-	-	07-08-2008	08-07-2017	setFormat-1	dates
2261993	-	-	93-22-1993	26-02-1993	setFormat-1	dates
8222000	-	-	22/02/2000	12/08/2000	setFormat-2	dates
11202001	-	-	20/12/2001	10/11/2001	setFormat-2	dates
casper	canal13	-	casper@canal13.cl	casper@canal13.com	generate-3	emails
caedgigo@garmendia.cl	-	-	gmail.com	garmendia.cl	getAfterAt-2	emails
1-845-456-7891	-	-	18454567819	18454567891	deletePunctuation-1	freetext
1-845-333-7891	-	-	18453337890	18453337891	deletePunctuation-1	freetext
1-7891	-	-	7891	17891	deletePunctuation-2	freetext
1-7892	-	-	7892	17892	deletePunctuation-2	freetext
Aliquam	-	-	a	A	firstCharacter-2	freetext
Computer and Communications Security	-	-	ACSAC	CCS	getCaps-2	freetext
Jones	1	-	Mr. Jones	Sr. Jones	addTitle-2	names
Dario Gag-Dorado	-	-	dari	daga	login-1	names
Paco Gabot Narale	-	-	paco	pagana	login-1	names
Daman Hivser-Kleiner	-	-	dakle	dahi	login-2	names
Giussepe Hindravtoks	-	-	giussepe	gihi	login-2	names
Agustino	Zimmann	-	Agustino, A.	Agustino, Z.	reduceName-5	names
618-4390	PAN	-	(7) 618-4390	(507) 618-4390	countryPrefix-3	phones
36-678-59-10	AUT	-	(9) 36-678-59-10	(43) 36-678-59-10	countryPrefix-7	phones
846-2730	AND	-	(846) 2730	(376) 846-2730	countryPrefix-7	phones
425-846-2730 425-425 425	-	-	425-425 425	425-846-2730	getNumber-2	phones
618 4390	425	-	425-618-4390	725-618-4390	setPrefix-1	phones
743-1650	425	-	425-743-1650	892-743-1650	setPrefix-1	phones
618-4390	425	-	(+425) 618-4390	(+725) 618-4390	setPrefix-5	phones
743-1650	425	-	(+425) 743-1650	(+892) 743-1650	setPrefix-5	phones
08	5	-	12	13	addTime-1	times
16:15:12	5	-	05:15:12	21:15:12	addTime-1	times
21:20	5	-	26:20	02:20	addTime-2	times
16:15:12	-	-	16:15:12:00	16:15:12	appendTime-1	times
16:15:12	30	-	16:15:12:30	16:15:12	appendTime-3	times
06:15:12	-	-	06:15:12	16:15:12	convert-1	times
08:40	EST	PST	20:40:00	05:40:00	convert-10	times
06:15	CET	MST	00:15:00	22:15:00	convert-10	times
14:10	12h	-	14:10	02:10 PM	convert-3	times
21:20	12h	-	21:20	09:20 PM	convert-3	times
08:40 UTC	24h	-	20:40	08:40	convert-4	times
06:15:12	24h	-	06:15:12	16:15:12	convert-4	times
14:10	-	-	14:10 AM	02:10 PM	convert-5	times
21:20	-	-	21:20 AM	09:20 PM	convert-5	times
08:40 UTC	-	-	03:40 PM	08:40 AM	convert-6	times
14:10	CET	UTC	14:10	13:10	convert-7	times
21:20	EST	UTC	03:20	02:20	convert-7	times
08:40	EST	PST	20:40	05:40	convert-8	times
06:15:12	CET	MST	18:15:12	22:15:12	convert-8	times
14:10	CET	UTC	14:10:00	13:10:00	convert-9	times
21:20	EST	UTC	07:20:00	02:20:00	convert-9	times
08	-	-	08	-	deleteTime-1	times
12.20dg	mg	-	12.200001	1220	convert-1	units
1854 dam	dm	-	18.540000	185400	convert-1	units
92.26 K	cK	-	92.26	9226	convert-2	units
12.2	dg	mg	12.200001	1220	convert-3	units
1854	dam	dm	18.540000	185400	convert-3	units
92.26	K	cK	92.26	9226	convert-4	units
81	hm	cm	0.081	8100	convert-4	units
1854 dam	-	-	Area	Length	getSystem-1	units
12.20dg	-	-	12.20	12.2	getValue-1	units

Table 6. Wrong answers provided by the GPT-3 DaVinci system in the 4-shot setting for examples based in large sentences.

		problem	**domain**
Input	Etiam dapibus tortor sed cursus blandit.		
Output	Etiamdapibustortorsedcursussed	deletePunctuation-3	freetext
Correct output	Etiamdapibustortorsedcursusblandit		
Input	Quisque mattis dolor ac metus fermentum consectetur.		
Output	Quismattisdoloracmetusfermentumconsectetur	deletePunctuation-3	freetext
Correct output	Quisquemattisdoloracmetusfermentumconsectetur		
Input	Integer malesuada elit et hendrerit fermentum.		
Output	Integermalesuadaelithendreritfermentum	deletePunctuation-4	freetext
Correct output	Integermalesuadaelitethendreritfermentum		
Input	Pellentesque tempor justo a dui varius maximus.		
Output	Pellentesque temporjustoaduivariusmaximus	deletePunctuation-4	freetext
Correct output	Pellentesquetemporjustoaduivariusmaximus		
Input	Mauris fringilla neque nec libero consequat quis facilisis nisi tristique		
Output	Maurisfringillanequeconsequatlibero	deleteSpaces-2	freetext
Correct output	Maurisfringillanequenecliberoconsequatquisfacilisisnisitristique		
Input	Nam cursus purus nec mi maximus fermentum		
Output	Namcursuspurusnecminimaximusfermentum	deleteSpaces-2	freetext
Correct output	Namcursuspurusnecmimaximusfermentum		
Input	Quisque mattis dolor ac metus fermentum consectetur.		
Output	QUISQUE MATTIS DOLOR AC METUS FERMENTUM CONSEQUETUR	toUpper-1	freetext
Correct output	QUISQUE MATTIS DOLOR AC METUS FERMENTUM CONSECTETUR		