Enhancing Semantic Resources via Large Language Models

Laura Ventrice[1,*], Giovanni Siragusa[1]

[1]Department of Computer Science, University of Turin, Italy

Abstract

Large Language Models (LLMs) have generated a great deal of interest in a wide range of domains. These models are trained on diverse and extensive text corpora, allowing them to learn patterns, relationships, and semantic understanding across various domains. By inputting prompts, the LLM can generate informative and contextually relevant responses, which can be used to solve a wide range of Natural Language Processing (NLP) tasks. One of the most relevant task in NLP is the representation of knowledge, where extracted information are organized according to a definite structure; however, the bottleneck of this operation is the time and effort required by annotators to organize them. For this reason, we present the idea of automatically populate a Knowledge Base using LLMs. We chose the Semagram one given its slot-filler structure which can be easily translated into prompts. We experimented on two LLMs, Vicuna and Mosaic Pretrained Transformers, with different output lengths, reporting interesting results.

Keywords

Knowledge Base, Large Language Model, Machine Learning, Neural Networks

1. Introduction

Large Language Models (LLMs) [1, 2, 3, 4, 5, 6] are becoming a relevant research trend. Their peculiar characteristic is the possibility of query them via a prompt, i.e. a text providing specific instructions. There are several methods to define the prompt: zero-shot prompt[7] where only the instruction is provided to the LLM; few-shot prompt[8] where some examples are provided to the model; and Chain-Of-Thoughts[9, 10] where the prompt (a zero-shot or a few-shot) is provided step-by-step in order to set the internal representation of the model on the task. The result of such prompts is a generated text.

A major goal of AI is knowledge representation, both to improve traditional NLP tasks (i.e. Word Sense Disambiguation, Machine Translation, etc.) and to enable the implementation of different applications (i.e. Question Answering, Information Retrieval, and so forth). There have been several attempts to encode knowledge, e.g. using Computational Lexicons such as WordNet[11] and broader resources like BabelNet[12], starting to define and use frames as in FrameNet[13]; other developing Corpus-based Models, mainly focused on the idea that similar words are used in similar contexts [14]. In order to solve most of the problems present in

GENERAL'23: GENerative, Explainable and Reasonable Artificial Learning Workshop 2023, held in conjunction with CHITALY 2023

*Corresponding author.

laura.ventrice@unito.it (L. Ventrice); giovanni.siragusa@unito.it (G. Siragusa)

0000-0002-1797-7956 (G. Siragusa)

existing approaches, the concept of semantic gram, or *Semagram*, was proposed in [15], creating a flexible structure in which each concept is encoded via a slot-filler structure. However, the major problem of these Knowledge Bases (KBs) is the effort necessary to build them because they require several domain experts and annotators over a long period of time.

In order to automatically scale the process of extending a KB, we thought that LLMs could be used to retrieve new concepts that could be possibly added. These models contain a vast knowledge, since they are trained over billions of textual data, and can be instructed to return new entries that satisfy a given condition via the prompt. Although any KB entry could be translated into a prompt, we decided to focus on the Semagram KB[15], given its slot-filler structure.

To the best of our knowledge, this is the first paper that tries to use the LLMs to populate a Knowledge Base. Our contribution is thus twofold:

1. We converted all $<$*slot*, *filler*$>$ pairs into a prompt for the Large Language Models. The mapping takes into account the category and the Part-Of-Speech tag of the filler to generate the input;
2. We experimented on two LLMs with 7 billions of parameters: Vicuna and Mosaic Pre-trained Transformer. For both models, we reported their Precision, Recall and F-Measure on different output lengths.

The remain of this paper is structured as follows: Section 2 contains the related works; Section 3 describes the Semagram Knowledge Base, reporting the description of each slot and some statistics; Section 4 describes how we defined the prompt and the problem we found in converting the slots; Section 5 reports the results on the two LLMs. Finally, Section 6 concludes the paper.

2. Related Works

Large Language Models. Large Language Models (LLMs), generally based on the Tranformer architecture [16], are pre-trained over massive unlabeled text. In the last years, they are becoming prevalent in the field of Natural Language Processing given their zero- and few-shot generalization ability. The most successful example is ChatGPT[6, 5], which made a wide impact on both AI research community and society. On the trend of ChatGPT, other reseach group and companies have released their open-source models, such as Bloom[1], Vicuna [2], Mosaic Pretrained Transformer [17] or LLaMA [3] among the others.

Knowledge Bases. Semagram[15] is based on the concept of semantic gram defined by Moerdijk et al. [18] to address the existing approaches to concept representation, providing a more formal description and a reduced number of slots (only 26 compared to the 200 of Moerdijk et al. [18]). It also include several features from other resources, such as Property Norms[19, 20] and the Visual Attributes[21]. Other resources that organize the lexical-semantic knowledge are: WordNet[11] and BabelNet[12] that provide a human-readable concept definitions; ConceptNet[22, 23] constructed from the MIT Open Mind Common Sense project[1] and regarding common-sense knowledge, i.e. shared and general facts or views of a set of concepts;

[1] https://www.media.mit.edu/projects/open-mind-common-sense/overview/

or FrameNet[13], which is based on the theory of frame defined by Fillmore et al. [24] where a meaning is encoded via simple slot-filler structure.

In the context of Knowledge Bases (KBs), our goal is to automatically expand the Semagram Knowledge Base using the Large Language Models. To the best of our knowledge, this is the first work that tries to use these models to accomplish such objective; other research papers used Open Information Extraction (OIE) methods[25, 26, 27] to create or expand KBs. For instance, Nakashole et al. [28] applied OIE to automatically build a taxonomy; Carlson et al. [29] and Speer and Havasi [30] used OIE to extend an existing ontology; and Siragusa et al. [31] extended the work of Cimiano and Wenderoth [32, 33] on the Qualia structure[34].

3. Semagram Knowledge Base

The Semagram Knowledge Base (KB)[15] is a flexible structure which encodes the semantic of a given concept via a slot-filler structure. The current version[2] is composed of 297 concepts and 26 slots; each concept (or filler) has a Babelnet[3] synset associated to it. Table 1 reports some statistics about the Semagram Knowledge Base, while Table 2 describes the 26 slots and their description.

It has been constructed by initially defining a set of 20 concepts over 10 categories: *animals*, *food*, *vehicles*, *clothes*, *home*, *appliance*, *instruments*, *artifacts*, *tools* and *containers*; then, the authors translated the concepts in Dutch and collected all the semagrams presented in the AWN dictionary [18], Property Norms [19, 20] and the Visual Attributes [21].

# concepts	297
# slots	26
# fillers	10194
avg. slots per concept	7.8 ± 1.89
avg. concepts per category	29.7 ± 0.46
avg. fillers per category	1019.4 ± 454

Table 1
The table reports some statistics about the Semagram Knowlege Base, such as the total number of concepts, the total number of fillers, the average number of concepts and fillers per category, and the average number of slots per concept.

4. The Prompt

Our aim in this paper is to automatically populate the Semagram KB using Large Language Models (LLMs) [2, 5, 4, 3] since they contain common knowledge. These models are based on the transformer architecture[16] and trained using billions of data; since they are generative models, it is possible to query them using a prompt, i.e. a sentence written in Natural Language expressing an instruction. Hence, our research objectives are the following:

[2]You can download it at: https://github.com/SapienzaNLP/Semagrams
[3]https://babelnet.org/

Slot	Description	Criteria
S1: accessory	All those objects that may have to do with X. The constraint is that there must be a physical contact and that the use of such object is strictly necessary for X.	`may have to do with {filler}`
S2: activity	All actions that X can actively or consciously do.	`can be {filler}`
S3: behavior	All the psychological features of X, including his attitude to his nature.	`can be {filler}`
S4: bodyPart	All the body parts which are involved in interacting with X.	`can have or be used with {filler}`
S5: colorPattern	All the features that refer to the color or texture of X.	`can be {filler}`
S6: consistency	All the entries with which the noticeable to the touch consistency or texture of X can be described.	`can be {filler}`
S7: content	All the entities which might be contained within X, without being constitutive parts of it.	`contain {filler}`
S8: efficiency	Positive (efficiency) or negative (ineff.) features of X related to his function.	`can be {filler}`
S9: generalization	Classification of X related to hypernyms.	`are {filler}`
S10: group	Names that indicates a group of animals of the same species of X.	`belong to {filler}`
S11: howToUse	All the actions or states required to operate, employ, interact with or perceive the existence of X.	Type of {filler}: • V or N: `can be used for {filler}` • A: `can be used when {filler}`
S12: material	Material of which X is composed.	`can be made of {filler}`
S13: movement	Terms that describe the type and speed of movement.	Type of {filler}: • N or V: `can {filler}` • A: `are {filler}`
S14: part	All the constitutive parts of X, with which it is normally presented.	`can have {filler}`
S15: place	All the entities in which X can be experienced, found or perceived.	`can be found or used in {filler}`
S16: product	All types of entity that can be derived from X through its processing or through natural processes.	`can be used to make {filler}`
S17: purpose	All of the purposes for which X is interacted with.	Type of {filler}: • V: `are used to {filler}` • N: `are used for {filler}`
S18: shape	Form of X.	`can be {filler}`
S19: size	Size of X.	`can be {filler}`
S20: smell	All the entries with which the smell of X can be described.	`can smell {filler}`

Table 2
The Semagram knowledge base model in terms of slots available with the corresponding description and the expression into which it is translated. The slots continue with Table 3 due to space limits.

Slot	Description	Criteria
S21: sound	All the entries with which the sound of X can be described.	Type of {filler}: • A: `sound {filler}` • V or N: `can {filler}`
S22: specialization	Classification of X in terms of his hyponyms.	`are {filler}`
S23: supply	The power mode that allows the functioning of X.	`use {filler}`
S24: taste	It contains information on the taste of a food.	`can be {filler}`
S25: time	All the entries which link X with the time flow or with specific moments of time.	Type of category: • Animals: `live during {filler}` • otherwise: `can be consumed or used during {filler}`
S26: user	All the kinds of living beings which are able to operate, employ, interact with or perceive X.	`are used by {filler}`

Table 3
Other slots that are included in the Semagram KB. Slots from 1 to 20 are described in Table 2.

RO1: How to convert a slot into a text for the prompt

RO2: How to define the prompt to query the LLM and retrieve new concepts for the Semagram KB

4.1. Converting the Slot

We started by analyzing the description of each slot in order to translate them into instructions. We found they follow simple ontology relations; for instance, the slot *material* can be translated in "*can be made of*". We thus decided to create a simple sentence that poses a criteria to the LLM. In this way, the model has to find all concepts that satisfy such constraint. Given the slot *howToUse*, we translated it in

```
can be used for {filler}
```

where the placeholder {filler} is a filler present in a category (e.g., *cut* for the category *food*).

For some slots, we discovered it is not possible to map them into a single criteria (e.g., *bodyPart*) because the filler does not fit with the passive form of the sentence. For this reason, we decided to write the condition as the disjunction of the active and passive form. Other slots, instead, require either the category of the concept or the Part-Of-Speech of the filler. Table 2 reports the defined criteria for each slot.

4.2. Creating The Prompt

Once the slots are converted, we created a simple and general prompt as follows:

```
Provide a list of 10 words that belong to the category and
satisfy the condition.
Desired output: comma separated list of words.
Category: {category}
Condition: {slot_criteria}
Output:
```

The first line contains the instruction for the model, i.e. it specifies what the model has to return and the constraints to satisfy, defined in "`Category`" and "`Condition`". The former contains a category of the Semagram Knowledge Base (e.g., animal); the latter reports the criteria of the selected slot and filler. We only used the slots and fillers present in the chosen category. The last line, "`Output:`", prioritizes the generation of the output by the model. Omitting this keyword will free the LLM to create any text before returning the concepts.

The resulting number of prompts is 3762.

5. Experiments

For our experiments, we chose Vicuna[2] and Mosaic Pretrained Transformer (MPT) [17] models. The former is an open-source Large Language Model (LLM) based on LLaMA[3] and fine-tuned on user-shared conversations; the latter is a decoder-style transformer fine-tuned for short-form instructions (i.e. MPT-instruct). For both the models, we used the version with 7 billions of parameters; we set temperature to 0.35, top probability to 0.7 and the number of beams to 4.

We tested 32, 64 and 128 tokens for the maximum output length of the model. We noticed that increasing the output length could improve the capability of the model to generate better answers. However, there were cases where a long output led to disruptive results; for instance, the model produced more bullet lists instead of comma separated ones. We also tested 256 tokens, finding that the output is similar to the 128 ones. The models were run on an NVidia RTX A6000, taking about 18 hours of computation to emit all the results.

We evaluated the models via Precision, Recall and F-Measure. Given $\mathbf{c}_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,n}\}$ as the set of concepts returned by the i-th prompt and $\mathbf{k} = \{k_1, k_2, \ldots, k_n\}$ as the set of concepts present in the Semagram Knowledge Base (KB), Precision is the ratio between the number of concepts generated by the model and present in the KB and $\mathbf{c}_i$:

$$Precision = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{|\mathbf{c}_i \cap \mathbf{k}|}{|\mathbf{c}_i|} \tag{1}$$

where $|P|$ is the total number of prompts.

The Recall is similar to the Precision function, with the difference that we divided by the number of concepts of the Semagram KB belonging to the prompt's category ($\mathbf{k}_{category}$):

$$Recall = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{|\mathbf{c}_i \cap \mathbf{k}|}{|\mathbf{k}_{category}|} \tag{2}$$

Finally, we calculated the F-Measure as the harmonic mean between Precision and Recall:

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

Notice that these measures give a partially insight of the performance of the prompt on a specific Large Language Model; their downside is that they are not able to evaluate the quality of the prompt (and the model) in returning new entries for the Semagram KB. In the future, we will perform a manual evaluation of the returned concepts.

5.1. Results

Table 4 reports Precision, Recall and F-Measure of Vicuna-7B and MPT-7B-Instruct model on the 3 output lengths. From the Table, we can see that the Vicuna-7B model has the highest F-Measure, resulting the best model. Furthermore, it is possible to see that both models performed well when the output length is set to 32 tokens. The increment of the output length (i.e., 64 and 128 tokens) caused instead a drop in the scores.

Model	Output length	P	R	F1
Vicuna-7B	32	0.074	0.321	0.102
	64	0.070	0.327	0.097
	128	0.069	0.327	0.097
MPT-7B	32	0.023	0.108	0.033
	64	0.007	0.051	0.011
	128	0.005	0.067	0.008

Table 4
Performance of Vicuna-7B and MPT-7B-Instruct with different output lengths. P stands for Precision, R for Recall and F1 for F-Measure.

We found that in some cases the models either returned no results or the output text was different from the requested one (i.e., a bullet list). We think this may be caused by either the prompt or the model (e.g., the number of parameters[4]). In order to give a quantitative idea of what we noticed, Table 5 reports the number of outputs in bullet-list form (not in comma-list form as requested) and the number of empty outputs (i.e., empty lines). From the table, it is possible to see that Vicuna-7B returned the desired output format more frequently than MPT-7B-Instruct; on the other hand, this latter one produced less empty lines. As future work, we are interested to investigate the reasons behind this phenomenon using different types of prompt (i.e. one-shot, few-shot or Chain-of-Thoughts) and LLMs.

Finally, we report in Table 6 the result of the condition "`are used by cooks`" for the *animal* category; we can notice that both models generated the word *dog*. This is another problem of

[4]More parameters mean a better LLM that is able to follow accurately the prompt instructions.

Model	Output length	# bullet list	# empty output
Vicuna-7B	32	341	246
	64	360	246
	128	359	246
MPT-7B	32	1495	9
	64	2234	9
	128	2096	9

Table 5
The table reports some insight analysis of the results produced by Vicuna-7B and MPT-7B-Instruct.

LLMs that we are interested to study, since biases and inaccuracies present in the training data can be propagated into the KB.

Model	Output length	List of concepts
Vicuna-7B	32	**dog**, chicken, beef, pork, lamb, fish, shrimp, lobster, crab, oyster
	64	**dog**, chicken, beef, lamb, pork, duck, goose, turkey, fish, shrimp
	128	**dog**, chicken, beef, lamb, pork, duck, goose, turkey, fish, shrimp
MPT-7B	32	**dog**, **cat**, cow, sheep, pig, chicken, turkey, duck, goose, ostrich
	64	**dog**, **cat**, cow, pig, sheep, chicken, turkey, duck, goose, ostrich
	128	**dog**, **cat**, cow, pig, sheep, chicken, turkey, duck, goose, ostrich

Table 6
The table reports the result for Vicuna-7B and and MPT-7B-Instruct of the prompt "`are used by cooks`". We can notice that both models generated the word *dog*, which is not factually accurate or representative of reality. MPT-7B-Instruct added also *cat* to the list.

6. Conclusions and future work

In this paper, we presented a preliminary work to automatically extend the Semagram Knowledge Base[15] using Large Language Models (LLMs). First, we defined the prompt for the LLMs mapping the $<$*slot*, *filler*$>$ pairs to a text-based constraint. Then, we evaluated two LLMs in returning concepts that are suitable for the Semagram KB. We used Vicuna and MPT-Instruct with 7 Billions of parameters for our experiments, discovering that the former one performed well on the task in terms of F-Measure. Furthermore, we found that increasing the output length has a negative impact on the performance of the model.

As future work, we are interested to explore how many new concepts, not present in the Semagram KB, are returned by the LLM, as well as different kinds of prompt (such as a few-shot or Chain-of-Thoughts prompt) and models. We will also investigate how the number of parameters of the LLM impacts on the quality of the results.

References

[1] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, arXiv preprint arXiv:2211.05100 (2022).

[2] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, et al., Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, See https://vicuna. lmsys. org (accessed 14 April 2023) (2023).

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[4] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al., Opt: Open pre-trained transformer language models, arXiv preprint arXiv:2205.01068 (2022).

[5] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[6] OpenAI, Introducing chatgpt., https://openai.com/blog/chatgpt, 2022.

[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, arXiv preprint arXiv:2109.01652 (2021).

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models, arXiv preprint arXiv:2201.11903 (2022).

[10] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, ArXiv abs/2205.11916 (2022).

[11] G. A. Miller, Wordnet: a lexical database for english, Communications of the ACM 38 (1995) 39–41.

[12] R. Navigli, S. P. Ponzetto, Babelnet: Building a very large multilingual semantic network, in: Proc. of ACL, Association for Computational Linguistics, 2010, pp. 216–225.

[13] C. F. Baker, C. J. Fillmore, J. B. Lowe, The berkeley framenet project, in: COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics, 1998.

[14] Z. S. Harris, Distributional structure, Word 10 (1954) 146–162.

[15] V. Leone, G. Siragusa, L. Di Caro, R. Navigli, Building semantic grams of human knowledge, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 2991–3000. URL: https://aclanthology.org/2020.lrec-1.366.

[16] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017.

[17] M. N. Team, Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL: www.mosaicml.com/blog/mpt-7b, accessed: 2023-03-28.

[18] F. Moerdijk, C. Tiberius, J. Niestadt, Accessing the anw dictionary, Information Systems (2008).

[19] K. McRae, G. S. Cree, M. S. Seidenberg, C. McNorgan, Semantic feature production norms

for a large set of living and nonliving things, Behav. r. m. 37 (2005) 547–559.
[20] B. J. Devereux, L. K. Tyler, J. Geertzen, B. Randall, The cslb concept property norms, Behavior research methods 46 (2014) 1119–1127.
[21] C. Silberer, V. Ferrari, M. Lapata, Models of semantic representation with visual attributes, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 572–582. URL: https://aclanthology.org/P13-1056.
[22] R. Speer, C. Havasi, J. CHAIDEZ, J. VENEZUELA, Y. KUO, Conceptnet 5, Tiny Transactions of Computer Science (2012).
[23] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.
[24] C. J. Fillmore, et al., Frame semantics, Cognitive linguistics: Basic readings 34 (2006) 373–400.
[25] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web., in: IJCAI, volume 7, 2007, pp. 2670–2676.
[26] F. Wu, D. S. Weld, Open information extraction using wikipedia, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 118–127.
[27] A. Moro, R. Navigli, Integrating syntactic and semantic analysis into the open information extraction paradigm., in: IJCAI, 2013.
[28] N. Nakashole, G. Weikum, F. Suchanek, Patty: a taxonomy of relational patterns with semantic types, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 1135–1145.
[29] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, T. M. Mitchell, Toward an architecture for never-ending language learning., in: AAAI, volume 5, 2010, p. 3.
[30] R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5., in: LREC, 2012, pp. 3679–3686.
[31] G. Siragusa, V. Leone, L. Di Caro, C. Schifanella, Discovering relational phrases for qualia roles through open information extraction, in: Knowledge Engineering and Semantic Web: 8th International Conference, KESW 2017, Szczecin, Poland, November 8-10, 2017, Proceedings 8, Springer, 2017, pp. 69–84.
[32] P. Cimiano, J. Wenderoth, Automatically learning qualia structures from the web, in: Proceedings of the ACL-SIGLEX workshop on deep lexical acquisition, Association for Computational Linguistics, 2005, pp. 28–37.
[33] P. Cimiano, J. Wenderoth, Automatic acquisition of ranked qualia structures from the web, in: ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007.
[34] J. Pustejovsky, The generative lexicon, Computational linguistics 17 (1991) 409–441.