

Leveraging LLMs for Semantic Type Annotation of Verbs' Arguments

Elisabetta Jezek, Gabriele Errico

Department of Humanities, University of Pavia

Piazza del Lino 2, 27100 Pavia, Italy

elisabetta.jezek@unipv.it, gabriele.errico01@universitadipavia.it

Abstract

In this paper, we present the results of two small-scale experiments aimed at verifying whether part of the Corpus Pattern Analysis (CPA) procedure developed by Hanks (2004) to manually extract recurrent language patterns from texts, can be automated using LLMs. Specifically, we examine ChatGPT and Gemini performance in the task of semantic type tagging of arguments in 150 Italian sentences realising 30 verb patterns (5 sentences per pattern). We run two experiments. In the first, we prompt ChatGPT to use the CPA ontology (about 200 hierarchically organized semantic types) in the annotation task; we provide the model with 5 sentences per pattern and ask it to assign the most specific type to the argument(s) of each sentence. In the second, we prompt both ChatGPT and Gemini to perform the task without the ontology, and ask the models to assign a single label to the argument(s) of the 5 sentences. Both experiments are performed in a zero-shot setting. We evaluate the results using the existing Italian T-PAS pattern resource as benchmark. Our results show that LLMs perform comparably well on both concrete and abstract type tagging and can therefore be used in a pilot study to support analysts in acquiring verb patterns from text.

Keywords: semantic type annotation, corpus pattern analysis, large language models

1. Background and Motivation

According to Hanks (2004, 2013), meaning is activated when words are combined with other words in patterns. To gain insight into the patterns of a language, one needs to study actual language behaviour as recorded in corpora. Hanks provides guidelines in the form of the Corpus Pattern Analysis (CPA) methodology for sorting and classifying language data. In CPA, typically, experts annotate a sample of about 250 concordance lines for a target verb, identifying recurrent patterns of verb usage and determining the semantic types of its arguments. Semantic types are selected from a hierarchically structured ontology (CPA ontology) of about 200 semantic types that has been developed for this purpose (Pustejovsky et al., 2006; Jezek and Hanks, 2010).

CPA is applied in practice in various languages, with the goal of building pattern inventories, e.g., English (Pattern Dictionary of English Verbs (PDEV), (Hanks and Pustejovsky, 2005), Spanish (Verbario, (Renau et al., 2019), Italian (Typed Predicate-Argument Structures (T-PAS), (Jezek et al., 2014); PhraseBase, (Giacomini and Rebosio, 2024), Croatian (CroaTPAS, (Marini and Jezek, 2019)) and Dutch (WoordCombinatie, (Colman and Tiberius, 2018)).¹ These resources are essential because by analyzing patterns in language, researchers may gain indirect access to how cognition is structured and operates. At the same time, with an increasing

proportion of texts generated by large language models (LLMs), studying authentic linguistic patterns is urgent. The dominance of English in LLM training data affects linguistic outputs in other languages, not only by introducing loan words but also by influencing grammar and discourse. This potential levelling and homogenization effect is further enhanced when AI-generated texts are again used as input for training the next generation of LLM.

Creating such inventories, however, is still mainly a computer-assisted, manual process and therefore very time-consuming. In this paper, we present the results of two small-scale experiments we conducted to explore whether LLMs could support certain aspects of the pattern-editing process. Building on previous experiments (Moretta et al., 2018), we specifically considered the models' performance in semantic type annotation of verbs' arguments. While semantic role labelling via LLMs is a widely performed task (Li et al., 2025), leveraging LLMs for semantic type tagging remains underrepresented.

It is important to note that a fundamental theoretical distinction exists between semantic roles and semantic types: a semantic type is an inherent attribute of an entity, while a semantic role has the attribute thrust upon it by the context (Jezek and Hanks, 2010). For example, a *knife* is inherently an [[Artifact]] which may play different roles depending on the context of use (e.g. *sharpen a knife* (Patient) vs *cut with a knife* (Instrument)). Despite the availability of different sets of tags for roles and types in lexical resources such as Framenet and VerbNet, roles and types are still often confused by annotators.

¹All these languages can be considered under-resourced in comparison to English.

In the following, we describe the experiments and report the results we obtained. The paper is organised as follows: in Section 2, we introduce the methodology we followed in our research and the resources we used as benchmarks for our experiments. In Sections 3 and 4, we present our experiments: we describe the experimental setup and report the results. In Section 5 we discuss our results. Section 6 concludes the paper.

2. Methodology

As referenced above, since the current CPA procedure is manual and labour-intensive, we examined whether an LLM could support some aspects of the pattern editing process. We specifically considered ChatGPT and Gemini, two of the most popular LLMs at present, and tested the models' performance in semantic type annotation of individual arguments. We devised two experiments: one in which ChatGPT is prompted to use the CPA ontology in the annotation task, and one in which both ChatGPT and Gemini are prompted without the ontology. For our experiment, we used a small selection of manually annotated data from the Italian T-PAS project as benchmarks (Gold Standard, GS) to evaluate the models' output. The resource includes 5,529 patterns for 1,164 Italian verbs and 252,943 corpus examples annotated with the verb pattern (<https://tpas.unipv.it>).

Based on related literature (Lakoff & Johnson, 1980; Han et al., 2023; FreedomIntelligence, 2023), we expected that semantic types related to concrete entities would be easier to assign than those related to abstract entities. To test this hypothesis, we constructed a dataset from the T-PAS data comprising 15 concrete semantic types and 15 abstract semantic types (see Table 1), each with varying levels of specificity/generalizability within the CPA hierarchy. For each type, we selected a verb together with 5 example sentences from one of its patterns, resulting in a dataset of 150 sentences in total.

All runs were performed through the LLMs' web interactive interface without API calls or fine tuning. In particular, Experiment 1 used ChatGPT-4o, while Experiment 2 used ChatGPT-5 and Gemini 1.5.

3. Experiment 1 (ChatGPT, with ontology)

The goal of the first experiment was to determine whether ChatGPT-4o could be used to annotate corpus data with semantic types from the hierarchically structured CPA ontology. We provided the model with the following information in the prompt: the role it should take on when performing the task (role-based prompting), some background on the project, and the comprehensive list of CPA types

inserted in their taxonomic structure. No examples were given; the prompting method was zero-shot. For each verb in Table 1, the model was asked to assign the most specific semantic type to the verb's arguments in each sentence, choosing from the types present in the ontology. The prompt can be found in the Appendix. For the statistical evaluation of the results of the first experiment, we decided to assign the scores as follows:

- 1 point for a perfect match, i.e., when the LLM returns as output the same label of the Gold Standard (corresponding to the label in the benchmark, column 2 in Table 1), for example, when the label of the Gold Standard is [[Water Vehicle]] and the LLM gives as output [[Water Vehicle]].
- 1 point for a partial match, i.e., when the LLM returns as output a label that is a direct subtype of the output of the Gold Standard, for example, when the label of the Gold Standard is [[Beverage]] and the LLM gives as output [[Alcoholic Drink]].
- 0 points for all the other cases, including non direct subtypes, for example when the label of the Gold Standard is [[Information]] and the LLM gives as output [[Abstract Entity]], because in the reference ontology ([[Information]]) is a subtype of [[Concept]] which is a subtype of [[Abstract Entity]].

The reason for assigning 1 point to both a perfect and a partial (more specific) match was that we explicitly instructed the model to choose the most specific semantic type for the argument in the individual sentences, and therefore felt it would be unfair to penalise the model for doing so. Using the above scoring method, we calculated accuracy, precision, recall, and F1 score using the Python scikit-learn library. We also created confusion matrices comparing the model's predicted labels with the Gold Standard labels to gain insight into the semantic types assigned by the model to arguments across sentences, and to compare these with those in the patterns. In the following subsections, we report the results we obtained.

3.1. Results for Concrete Types

For the annotation of concrete semantic types, we achieved a high accuracy (0.75, see Table 2). If we consider the weighted average, i.e., a mean that takes into account the relative importance (or weight) of each data point in both experiments, the precision is higher than the recall. This means that the number of false positives is lower than the number of false negatives: ChatGPT-4o is more likely to miss the correct label (false negatives) than to assign the wrong one (false positives).

TopType	SemType	Verb-IT	Translation	SynRole
Concrete	Alcoholic_Drink	trincare	swallow down	obj
Concrete	Water_Vehicle	affondare	sink	sub
Concrete	Flying_Vehicle	atterrare	land	sub
Concrete	Cloth	ricamare	embroider	obj
Concrete	Bomb	esplodere	explode	sub
Concrete	Vehicle	parcheggiare	park	obj
Concrete	Musical_Instrument	accordare	tune	obj
Concrete	Beverage	bere	drink	obj
Concrete	Garment	indossare	wear	obj
Concrete	Device	scattare	shoot	sub
Concrete	Food	mangiare	eat	obj
Concrete	Flag	sventolare	wave	sub
Concrete	Artifact	fabbricare	make	obj
Concrete	Human	addomesticare	tame	sub
Concrete	Inanimate	brillare	shine	sub
Abstract	Software	costruire	build	obj
Abstract	Document	scrivere	write	obj
Abstract	Musical_Composition	comporre	compose	obj
Abstract	Business_Enterprise	avviare	start	obj
Abstract	Emotion	annegare	drown	obj
Abstract	Information	diffondere	spread	obj
Abstract	Goal	raggiungere	reach	obj
Abstract	Deity	adorare	adore	obj
Abstract	Money	pagare	pay	obj
Abstract	Time Period	passare	pass	sub
Abstract	Time Point	suonare	sound	sub
Abstract	Concept	sposare	marry	sub
Abstract	Responsibility	addossare	put on	obj
Abstract	Opportunity	sprecare	waste	obj
Abstract	Abstract_Entity	cancellare	delete	obj

Table 1: Dataset of Italian verbs with semantic types and syntactic roles

	precision	recall	f1-score	support
accuracy			0.75	75
macro avg	0.52	0.45	0.46	75
weighted avg	0.87	0.75	0.76	75

Table 2: Evaluation of Concrete Semantic Types

In Figure 1, we report the confusion matrix for the annotations of concrete types. As mentioned above, we compare the labels predicted by GPT-4o (“predicted labels” on the horizontal axis) with the “true labels” assigned by the human annotators in T-PAS (vertical axis) to gain insight into the semantic types the model assigns to arguments in individual sentences. The matrices use a gradient colour scale to represent frequency counts. In this case, the lighter the color, the lower the count in the cell; conversely, the darker the color, the higher the count. The cell with the darkest colour corresponds to 5 correct instances (which is the number of sentences we asked the model to process for each semantic type). For example, as shown in Fig. 1 [[Water Vehicle]] was always correctly annotated, and the corresponding cells in the matrices are dark. By contrast, [[Beverage]] was labelled as

such only once, which is reflected by the pale colour of the corresponding cell in the confusion matrix. The objects in three of the five example sentences were labelled twice as [[Water]], once as [[Liquid]] and once as [[Alcoholic Drink]]. An intermediate example is [[Food]]: this semantic type, which is a complex type in the ontology (i.e., a type made up of two components, i.e. [[Meal [Food, Activity]]], was labeled twice as [[Meal]], once as [[Animal]], once as [[Fish]], and only once as [[Food]] (see below for further comments on this annotation).

For the 15 concrete types that we tested, GPT-4o assigned 29 different labels. In addition to the 10 concrete types in Table 1, the following labels appear: [[Animal]], [[Fire]], [[Firearm]], [[Fish]], [[Furniture]], [[Human Group]], [[Light]], [[Light Source]], [[Liquid]], [[Meal]], [[Road Vehicle]][[System]], [[Water]], [[Weapon]].

As referenced above, in the prompt we asked the model to identify the most specific type for a lexical item in particular syntactic positions within a sentence (subject or object). This proved to be highly effective with verbs that select very specific semantic types in the pattern slots, as in the case of [[Water Vehicle]], [[Alcoholic Drink]], [[Bomb]], [[Flag]], [[Flying Vehicle]], [[Garment]] and [[Musical

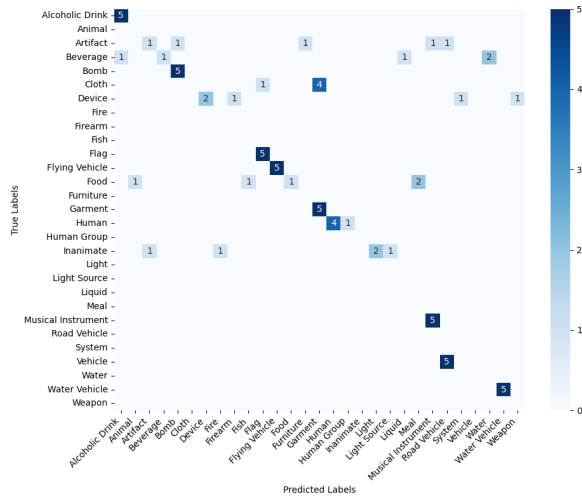


Figure 1: Confusion Matrix for Concrete Types (Experiment 1)

cal Instrument]]. On the other hand, some verbs do not select very specific semantic types in the pattern slots. For example, the verb *bere* 'to drink' does not select only one specific beverage, but rather all kinds of beverages: [[Beverage]] is the gold standard for the pattern. Considering that in the prompt we explicitly asked for the most specific type, it is only to be expected that the model would identify the type [[Water]] in cases where a sentence contains the word *acqua* 'water'. In the sentence *E' fondamentale ricordarsi di non bere acqua corrente* 'It is essential to remember not to drink tap water' the LLM labelled the object *acqua corrente* ('tap water') as [[Water]].

Similarly, the verb *fabbricare* 'to build, to make' can apply to several types of objects. The gold standard to capture this generality is the general type [[Artifact]], and the model identified more specific types in the different sentence, such as [[Furniture]] *mobili per ingresso*, [[Road Vehicle]] *vagone di acciaio*, [[Musical Instrument]] *strumento musicale*, which are all subtypes of [[Artifact]].

This raises an interesting issue about the relationship between the words in the texts and the types in the patterns (Jezek & Hanks, 2010): the type in the pattern is a generalisation over the word in the text. In a future experiment, it may therefore be of interest to include a further instruction in the prompt and ask the model to identify not only 'the most specific semantic type' but also 'the most specific semantic type that matches all the sentences provided'. This would also prove the ability of the model to generalize. For now, we adjusted the scoring method so that instances where the model chose a more specific type would not negatively affect the overall model's performance (see Table 2).

Polysemous lexical items are difficult for the LLM

to identify. We identified cases of regular polysemy involving some of the objects of the Italian verb *mangiare* ('to eat'). In the sentence *Io proprio non ce la faccio a mangiare un animale, è più forte di me* 'I just can't eat an animal — I can't help it, it's just the way I am', the object *un animale* 'an animal' is assigned the semantic type [[Food]] in T-PAS, whereas the LLM assigns the label [[Animal]]. A similar case is found with the object *pesce freschissimo* 'incredibly fresh fish' in the sentence *Ti consiglio qualsiasi ristorante nel Barrio Alto dove potrai mangiare pesce freschissimo a poco prezzo* 'I recommend any little restaurant in Bairro Alto, where you can eat incredibly fresh fish at a low price': while in T-PAS it is labelled as [[Food]], the LLM assigns the label [[Fish]]. *Animale* 'animal', like its hyponym *pesce* 'fish', is a polysemous word, as it can refer both to the living animal and to the dead one intended for consumption. Since this semantic relationship is currently not represented in the ontology, both cases received the score of 0. However, future implementations of the resource and the ontology could take this polysemous relationship into account.

Another issue is that some semantic types are interpreted differently by the LLM than they are intended in the ontology. For example, in the ontology, [[Human group]] is intended only for collective nouns, such as *family*, *class* or *police*. The LLM, by contrast, uses the label [[Human Group]] for everything that is 'more than one [Human]', assigning this label to plural nouns and pronouns, such as *they*. As a consequence, the matrix scores for [[Human]] and [[Human Group]] are lower (see Figure 1). For example, *gli uomini* ('the men'), the subject of the sentence *Gli uomini sapevano coltivare il grano e addomesticavano gli animali* ('The men knew how to cultivate wheat and domesticated animals'), was labelled as [[Human Group]].

3.2. Results for Abstract Types

As with the concrete types, the accuracy score for the annotation of the 15 abstract types is high: 0.79. In addition, as shown in Table 4, the precision score (weighted avg. Italian 0.81) is higher than the recall score (weighted avg. 0.79). The initial observation is that GPT-4o performed slightly better with abstract entities compared to concrete ones. On the one hand, this contradicts other studies: Han et al. (2023) and FreedomIntelligence (2023) state that ChatGPT shows promising skills in recognising concrete entities, while recognising abstract entities remains more challenging. This is probably because concrete entities are usually associated with more clearly defined sensory representations and richer contextual information (Peeters et al. 2023). On the other hand, we suggest that the level of polysemy for abstract types is lower. Moreover, the

level of correspondence between lexical items and types is higher. For example, in the sentence *Non sprechiamo questa opportunità di modificare una legge dannosa sia dal punto di vista fisico che psicologico* ('Let's not waste this opportunity to amend a law that is harmful both physically and psychologically'), there is a strong match between opportunità ('opportunity') and the label [[Opportunity]]. The lower number of abstract types in the CPA ontology compared to concrete types (i.e., 42 semantic types under [[Abstract Entity]] vs. 101 under [[Physical Entity]]) may also contribute to the higher scores.

	precision	recall	f1-score	support
accuracy			0.79	75
macro avg	0.58	0.56	0.56	75
weighted avg	0.81	0.79	0.79	75

Table 3: Evaluation of Abstract Semantic Types

Considering the confusion matrix in Figures 2, we observe that there are far fewer semantic types in the output for the abstract types than for the concrete types, i.e., only 16 compared to 26. The matrix also shows a higher number of dark cells compared to the confusion matrix for the concrete types. For the abstract types [[Business Enterprise]], [[Information]], [[Opportunity]], [[Responsibility]], and [[Time Period]], ChatGPT carried out the annotation completely correctly. At the same time, the number of pale cells is lower.

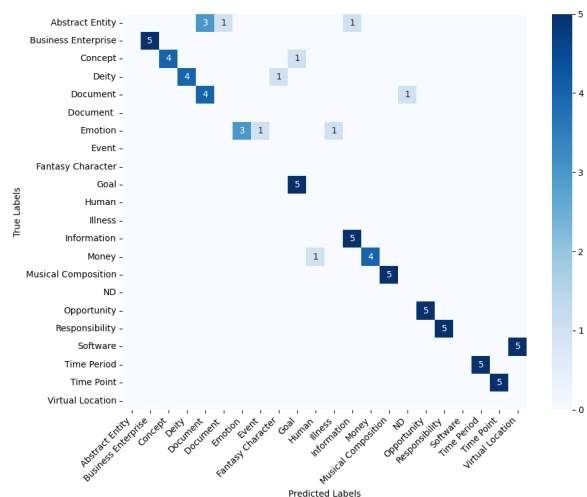


Figure 2: Confusion Matrix for Abstract Types, (Experiment 1)

Some semantic types, such as [[Software]], are challenging to annotate for both humans and LLMs. ChatGPT assigned the label [[Virtual Location]] to all five example sentences where the Gold Standard indicated [[Software]].

4. Experiment 2 (ChatGPT, no ontology)

Experiment 2 tests whether semantic type annotation can be performed without explicitly including the CPA ontology in the prompt. The dataset, interface, and workflow match those of Experiment 1 (interactive chat environment), so that the results remain directly comparable. However, it is important to note that the evaluation protocols differ across experiments. In Experiment 1, both perfect matches and direct subtype matches with the Gold Standard are scored as correct. By contrast, in Experiment 2, only perfect matches and accepted near-synonyms receive a correct score. Other outputs, including labels that are too general or too specific, are scored as incorrect. In Experiment 2, the model is then asked to identify the relevant argument(s) and assign the most specific semantic type that can extend across the full five-sentence set for the same verb slot.

4.1. Results for Concrete Types

Removing the ontology from the prompt led to a clear improvement in concrete-type annotation accuracy, increasing it to 0.80. These findings contrast with those from ontology-based approaches (Ouyang et al., 2023) but align with the notion of the predictable unpredictability of LLMs output, suggesting that prompt design is part of the method for annotating in the social sciences (Atreja et al., 2024). At the aggregate level, precision along with recall are balanced in this configuration. Figure 3 provides class-level evidence on remaining mismatches.

	precision	recall	f1-score	support
accuracy			0.80	75
macro avg	0.67	0.67	0.67	75
weighted avg	0.80	0.80	0.80	75

Table 4: Evaluation of Concrete Semantic Types

In Figure 3 we report the confusion matrix for concrete types in Experiment 2. The vertical axis contains the true labels from the GS, while the horizontal axis contains the adjudicated labels from the Score column (not predictions seen in Experiment 1). In this way rather than a gradient of blues the matrix shows either a complete alignment or a mismatch.

As we have seen in Experiment 1 for [[Beverage]] and for [[Food]], predictions included granularity shifts and polysemy effects (e.g., food-as-edible-entity vs animal-as-living-entity). Model errors on fine-grained hierarchy levels and no ontology prompt may help reach higher aggregates

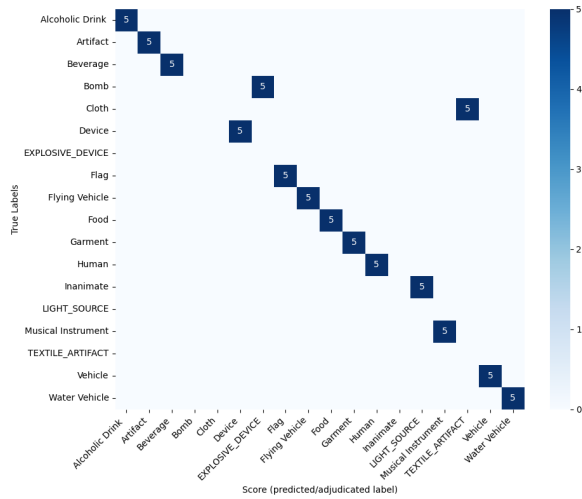


Figure 3: Confusion Matrix for Concrete Types (Experiment 2, ChatGPT)

scores on semantic concrete types (see Section 3.1, accuracy 0.80 vs 0.75; macro F1 0.67 vs 0.46). Near-synonyms were accepted (e.g., Air Vehicle/Space Vehicle for [[Flying Vehicle]]), whereas super/sub typer were rejected (e.g., Textile Artifact for [[Cloth]]).

4.2. Results for Abstract Types

For abstract semantic types in Experiment 2 (ChatGPT, no ontology), accuracy is 0.73, with macro-average precision/recall/F1 at 0.58 and weighted precision/recall/F1 at 0.73. Similar results have also been demonstrated in legal studies (Savelka, 2023). At the class level, the error profile confirms that semantically related but over-general labels are penalized in the adjudicated Score: for example, Artifact / Digital Artifact for [[Software]] is scored as incorrect, and Music for [[Musical Composition]] is also scored as incorrect. As mentioned, only near-synonymous were retained, such as Text for [[Document]] and Organization / Business Activity for [[Business Enterprise]]. Figure 4 reports the confusion matrix for abstract types.

	precision	recall	f1-score	support
accuracy			0.73	75
macro avg	0.58	0.58	0.58	75
weighted avg	0.73	0.73	0.73	75

Table 5: Evaluation of Abstract Semantic Types

In Experiment 1, abstract types slightly outperformed concretes. In Experiment 2, the direction partly reverses (0.73 vs 0.79), while macro F1 slightly rises (0.58 vs 0.56), opening an interesting conversation about abstractness and specificity (Villani et al., 2024). A plausible interpretation, con-

sistent with the same logic, is that removing the ontology and requiring a single label across five sentences increases pressure toward broader labels, especially in abstract domains (e.g., Artifact/Digital Artifact for [[Software]], Music for [[Musical Composition]]).

Concrete types, by contrast, remain more strongly anchored to sentence-level lexical evidence and therefore retain higher aggregate performance.

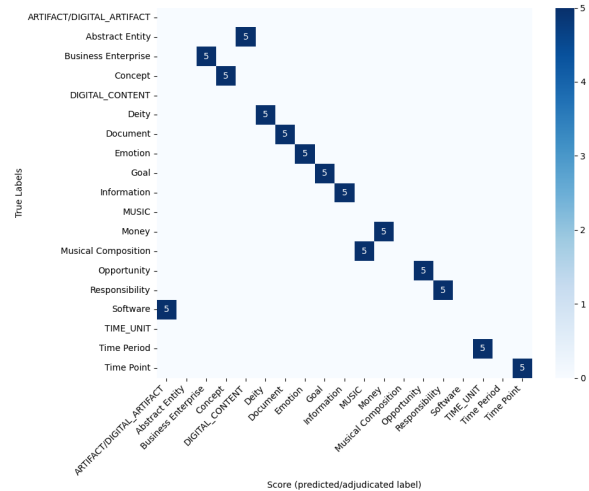


Figure 4: Confusion Matrix for Abstract Types (Experiment 2, ChatGPT)

5. Experiment 2 (Gemini, no ontology)

This parallel run mirrors Experiment 2 (ChatGPT) using the same dataset and no-ontology prompt objective. Predictions are logged in the same spreadsheet format and scored according to the Experiment 2 criteria described above.

5.1. Results for Concrete Types

	precision	recall	f1-score	support
accuracy			0.73	75
macro avg	0.61	0.61	0.61	75
weighted avg	0.73	0.73	0.73	75

Table 6: Evaluation of Concrete Semantic Types

For concrete semantic types the model reaches an accuracy of 0.73, with the other metrics at 0.61 and weighted averages at 0.73. As with the Experiment 2 (ChatGPT, no ontology), residual errors are mainly linked to label granularity. For example, Beverage was assigned where the GS was [[Alcoholic Drink]], too vague, so incorrect; by contrast, Aircraft for [[Flying Vehicle]] was retained acceptable

as considered a near synonym. A further mismatch concerns cases where [[Inanimate]] was rendered Light Source, i.e., very specific, so, rejected.

Compared with Experiment 2 (ChatGPT, no ontology), Gemini performs worse (accuracy 0.73 vs 0.80; macro F1 0.61 vs 0.67). This pattern is confirmed by the comparison against ChatGPT ontology: Gemini shows slightly lower accuracy (0.73 vs 0.75), even if it improves on macro F1 (0.61 vs 0.46). This pattern suggests a more balanced class-level behavior. Figure 5 reports the confusion matrix for Gemini concrete types, with Gold Standard labels on the Y-axis and adjudicated labels Score on the X-axis.

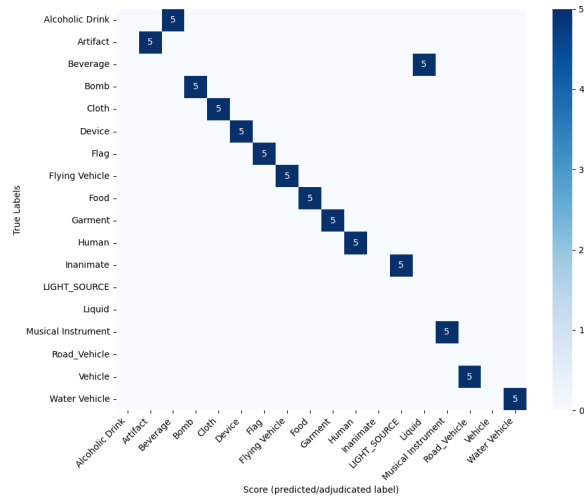


Figure 5: Confusion Matrix for Concrete Types (Experiment 2, Gemini)

5.2. Results for Abstract Types

For abstract semantic types in Experiment 2 (Gemini, no ontology), accuracy is 0.73, with macro-average precision/recall/F1 at 0.58 and weighted precision/recall/F1 at 0.73. [[Software]] was correctly captured, as well as Point in Time for [[Time Point]], accepted as near-synonym, troubling for ChatGPT; by contrast, Music for [[Musical Composition]], Psychological Feature for [[Emotion]], and Abstract Entity for [[Concept]] are plausibly incorrect with the current task.

Compared with Experiment 2 (ChatGPT, no ontology), aggregate indices are aligned (accuracy 0.73 and macro F1 0.58), with homogeneous performance overall and heterogeneous errors in different specific classes. Compared with Experiment 1 (ChatGPT, with ontology), abstract-type accuracy is lower in Gemini Experiment 2 (0.73 vs 0.79), while macro F1 is slightly higher (0.58 vs 0.56). These outcomes are consistent with the most recent research in the field. As above, this cross-experiment reading ought to remain careful because the scor-

ing policies are not identical. Figure 6 presents the confusion matrix for Gemini abstract types, with Gold Standard labels on the Y-axis and adjudicated labels Score on the X-axis.

	precision	recall	f1-score	support
accuracy			0.73	75
macro avg	0.58	0.58	0.58	75
weighted avg	0.73	0.73	0.73	75

Table 7: Evaluation of Abstract Semantic Types

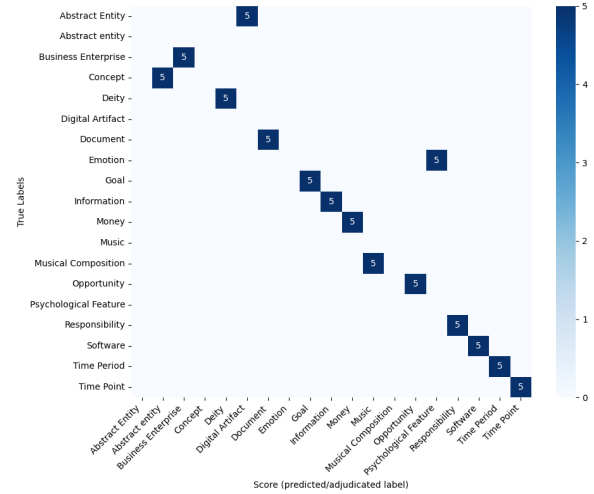


Figure 6: Confusion Matrix for Abstract Types (Experiment 2, Gemini)

Considering the confusion matrices in Figures 5 and 6, we observe that there are far fewer semantic types in the output for the abstract types than for the concrete types, i.e., only 16 compared to 26. The matrix also shows a higher number of dark cells compared to the confusion matrix for the concrete types. For the abstract types [[Business Enterprise]], [[Information]], [[Opportunity]], [[Responsibility]], and [[Time Period]], the model carried out the annotation completely correctly.

Some concepts are difficult to annotate, both for humans and for LLMs. The semantic type [[Software]] proved difficult. The model assigned the label [[Virtual Location]] to all 5 example sentences whereas the Gold Standard assigned [[Software]].

6. Discussion

From the Delta Snapshots in Table 9, we can notice that the contrast in concretes is greater, while abstracts are more aligned. In Table 8, under the tested settings, we can see a moderate split in semantic domains: for concretes (C-Acc) the accuracy metric puts ChatGPT without ontology at the top of the ranking, followed by its counterpart with

ontology, Gemini stays behind (0.73). For Abstracts (A-Acc) ChatGPT with ontology is the highest, while no ontologies setups converge (both 0.73). Delta snapshots are provided in Table 9 for visual aid of the differences across setups.

From these results, a reasonable and cautious explanation might be that prompting the LLM with packed information, such as ontologies, constrains the model on concrete labeling while stabilizing boundaries in abstract categorization. This is probably because the text of the ontology itself contains vectorial information that drives the model to assign an incorrect label due to the high pressure and high speed during the task, thereby 'distracting' the model from assigning the correct one. In this way, no ontology prompting scores higher on concretes assignments because lexical evidence, especially in stronger lexical binds across sentence sets occur (see, for instance, `[[Alcoholic Drink]]`) does drive the model toward a less 'noisy' annotation task.

	C-Acc	C-F1	A-Acc	A-F1
GPT1	0.75	0.46	0.79	0.56
GPT2	0.80	0.67	0.73	0.58
Gem	0.73	0.61	0.73	0.58

Table 8: Summary Evaluation

	C-Acc Δ	C-F1 Δ	A-Acc Δ	A-F1 Δ
GPT2-GPT1	+0.05	+0.21	-0.06	+0.02
Gem-GPT1	-0.02	+0.15	-0.06	+0.02
GPT2-Gem	+0.07	+0.06	0.00	0.00

Table 9: Delta Snapshots across Experiments

Beyond the differences in aggregate scores, Gemini remains methodologically useful due to the fact that it provides a very user friendly explanation, guiding the user through why a label was selected or not (see Appendix c).

Taken into account that in Experiment 2 the models were not given any piece of information about the ontology, the results are quite astounding, they basically came up with their own types, which are, as shown by our research, directly comparable to the ones in the CPA ontology. The model didn't see the information, was not trained, and the possibilities that these models were trained through a specific fine-tuning for semantic types tagging, in respect to Hanks' ontology, are scarce.

7. Concluding remarks and further research

In this paper, we present the results of two small-scale experiments we conducted to verify whether

part of the CPA procedure utilized to identify Italian verb patterns in corpora can be automated. In particular, we explore the performance of ChatGPT and Gemini on the task of semantic type annotation of verb arguments, using the existing T-PAS resource as benchmark. We set up the experiments to test the model's performance on both concrete and abstract semantic types. Overall, the results show that LLMs perform comparably well on both concrete and abstract type tagging.

From a practical standpoint, these results suggest that resource builders might markedly streamline their annotation workflow by incorporating LLM suggestions as a preliminary step. Similar workflows have also been implemented in historical studies (Celli and Mingazov, 2024). For example, a lexicographer might substantially reduce the time required for manual annotation by using an LLM to pre-annotate arguments' types. Then thereby, this allows to shift human expertise toward verification and correction of the model-proposed predictions. This approach could free up time for further linguistic analysis. In practical use cases, such as building computational lexicons, updating language resources, or assisting large-scale corpus studies, this workflow adjustment could improve scalability and budget efficiency, particularly when expert human resources are scarce or swift data processing is required. Remember that the interface and the models used were the free versions accessible to everyone, not giving any particular prompt-engineered advantage (Gilardi et al., 2023). Even small models can be helpful, from a mediocre resource to a better one (Simonetti et al., 2024). Considering this, prompt adjustment can sometimes take an unpredictable amount of time to measure, depending on the task.

Finally, human are better in performing label abstraction, but LLMs can be surprisingly accurate. Humans can be subject to tiredness, while LLMs have other constraints. For instance humans can directly infer that the verb to drink (it. *bere*) requires a general pattern label `[[Beverage]]`. This might denote a recurring mismatch between specific task a consistent generalization pattern. Polysemy, as seen with the difficulty to disambiguate between `[[Animal]]` (a living creature) instead of `[[Food]]` and group identification, as seen with plural label tagging resulting in generalization (collective nouns have not been contemplated by the model), clearly show that LLMs tagging should be carefully analyzed and realigned.

Ongoing research includes investigating whether LLMs are accurate in assigning these labels in other languages with CPA inventories: we already run preliminary experiments in Dutch with satisfying results. Future research includes: trying to use an open-source LLM such as Llama for the experi-

ments; expanding the dataset using more T-PAS data with the ultimate goal of developing an LLM-based semantic type labeler; studying the extent to which automatically pre-annotating sentences for patterns affects final annotation, by comparing LLMs assisted annotation with manual annotation from T-PAS data.

8. Acknowledgements

We are thankful to four reviewers for their insightful comments on an earlier version of this paper. Elisabetta Jezek has been partially funded by the University of Pavia with the ENHANCE SH project.

9. Limitations

The dataset we used is arguably a small dataset, and would benefit from expansion. While we considered existing resources and extracted linguistic data with the desired features from those, future work could expand it by collecting new data via human annotation, generation, or other data-driven approaches.

The authors initially performed multiple runs of Experiment 1 to observe the 'predictable unpredictability' of the models. Although, during the design of Experiment 2, a single-output was preferred to simulate a high-speed, efficient work scenario, instead of evaluating the unpredictability of outputs.

These results are considered very valuable by the authors and could possibly help not only in being less biasing during a human-in-the-loop pipeline work; or model-in-the-loop where humans construct the GS and the model are requested, without seeing the data first, to annotate semantic types, and only in the end human variation evaluation comparison checkups (GS vs. LLMs' output); but also address a variability that otherwise stays unchecked.

A further limitation interest hyperparameter setting: because the experiments were handled via web interfaces temperature was neither retrievable nor configurable. This binds pragmatically to our ready for use approach. However the authors are aware that unknowability may affect reproducibility.

While the present work performs an in-depth evaluation of how LLMs behave when faced with semantic typing, our research does not explore the inner mechanisms that underlie this capability. We acknowledge that doing so would provide complementary evidence that may be needed to shed full light on the phenomenon. Moreover, research could focus on how LLMs handle semantic typing in more naturalistic scenarios, e.g., in the context of real-world NLP applications, which is something the current work does not explore.

10. Ethical considerations

While this work presents no serious ethical concerns, a general consideration needs to be made about the use of pre-trained LLMs. As is commonly acknowledged, these models should be used with caution as they could perpetuate harmful biases present in their training data. Furthermore, there is a risk that they will generate false or misleading output. In our work, we minimize these risks as we do not use the LLMs to generate output, but only to score the plausibility of sentences fed as input.

11. Bibliographical References

- S. Atreja, J. Ashkinaze, L. Li, J. Mendelsohn, and L. Hemphill. 2024. [Prompt design matters for computational social science tasks but in unpredictable ways](https://doi.org/10.48550/arXiv.2406.11980). *arXiv preprint arXiv:2406.11980*. <https://doi.org/10.48550/arXiv.2406.11980>.
- F. Celli and D. Mingazov. 2024. [Knowledge extraction from LLMs for scalable historical data annotation](https://doi.org/10.3390/electronics13244990). *Electronics*, 13(24). <https://doi.org/10.3390/electronics13244990>.
- L. Colman and C. Tiberius. 2018. A good match: A dutch collocation, idiom and pattern dictionary combined. In *Proceedings of the XVIII EURALEX International Congress: Lexicography in global contexts*, pages 233–246, Ljubljana, Slovenia.
- L. Giacomini and L. Rebosio. 2024. Introducing PhraseBase: A linguistic information system for language learners and for NLP. In *PhrasaLex III Workshop on Phraseological Approaches to Lexicography*, Innsbruck, Austria. Conference presentation.
- F. Gilardi, M. Alizadeh, and M. Kubli. 2023. [ChatGPT outperforms crowd-workers for text-annotation tasks](https://doi.org/10.48550/arXiv.2303.15056). *arXiv*. <https://doi.org/10.48550/arXiv.2303.15056>.
- P. Hanks. 2004. Corpus pattern analysis. In *Proceedings of the 11th EURALEX International Congress*, volume 1, pages 87–97. Université de Bretagne-Sud.
- P. Hanks. 2013. *Lexical analysis: Norms and exploitations*. MIT Press, Cambridge, MA.
- P. Hanks and J. Pustejovsky. 2005. A pattern dictionary for natural language processing. *Revue Française de Linguistique Appliquée*, 10(2):63–82.
- E. Jezek and P. Hanks. 2010. What lexical sets tell us about conceptual categories. *LEXIS*, 4:7–22.

- E. Jezek, B. Magnini, A. Feltracco, A. Bianchini, and O. Popescu. 2014. T-PAS: A resource of typed predicate argument structures for linguistic analysis and semantic processing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 890–895, Paris. European Language Resources Association (ELRA).
- Xinxin Li, Huiyao Chen, Chengjun Liu, Jing Li, Meishan Zhang, Jun Yu, and Min Zhang. 2025. [LLMs can also do well! breaking barriers in semantic role labeling via large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23162–23180, Vienna, Austria. Association for Computational Linguistics.
- C. Marini and E. Jezek. 2019. CroatPAS: A resource of corpus-derived typed predicate–argument structures for croatian. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, Torino. Accademia University Press.
- F. Della Moretta, A. Feltracco, E. Jezek, and B. Magnini. 2018. Designing a methodology for semantic type tagging of argument positions. *Italian Journal of Computational Linguistics*, 4(2):1–16.
- S. Ouyang, J. Huang, P. Pillai, Y. Zhang, Y. Zhang, and J. Han. 2023. [Ontology enrichment for effective fine-grained entity typing](#). *arXiv preprint*. <https://doi.org/10.48550/arXiv.2310.07795>.
- J. Pustejovsky, C. Havasi, J. Littman, A. Rumshisky, and M. Verhagen. 2006. Towards a generative lexical resource: The brandeis semantic ontology. In *Proceedings of LREC*, pages 1702–1705. European Language Resources Association (ELRA).
- I. Renau, R. Nazar, A. Castro, B. López, and J. Obreque. 2019. Verbo y contexto de uso: Un análisis basado en corpus con métodos cualitativos y cuantitativos. *Revista Signos*, 52(101):878–901.
- J. Savelka. 2023. [Unlocking practical applications in legal domain: Evaluation of GPT for zero-shot semantic annotation of legal texts](#). *arXiv preprint arXiv:2305.04417*. <https://doi.org/10.48550/arXiv.2305.04417>.
- L. Simonetti, E. Jezek, and G. Vetere. 2024. Subcategorization of italian verbs with LLMs and T-PAS. In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*. <https://aclanthology.org/2024.clicit-1.99.pdf>.
- C. Villani, A. Loia, and M. M. Bolognesi. 2024. [The semantic content of concrete, abstract, specific, and generic concepts](#). *Language and Cognition*, 16(4):867–894. Published online by Cambridge University Press on 18 January 2024. <https://doi.org/10.1017/langcog.2023.64>.

12. Appendix

12.1. Appendix A: Prompt Experiment 1 ChatGPT

Sei un linguista che sta lavorando per la risorsa T-PAS, un archivio di Typed Predicate Argument Structures, (anche dette pattern) per l'italiano. Tali pattern sono stati acquisiti manualmente da corpora attraverso l'analisi delle informazioni distribuzionali dei verbi italiani, seguendo la metodologia della Corpus Pattern Analysis descritta in Hanks (2004, 2013). Per ogni verbo presente nella risorsa, T-PAS mostra quali sono i suoi pattern più ricorrenti. I pattern sono costituiti dal verbo e dai suoi argomenti (soggetto, oggetto, complemento preposizionale, ecc.), a ognuno dei quali è assegnata una etichetta che ne identifica le proprietà semantiche (dette tipo semantico, ad esempio HUMAN, ARTIFACT, EVENT, LOCATION, ecc.). Per l'annotazione dei tipi semantici, gli annotatori si sono serviti di una ontologia, cioè di un elenco di etichette, organizzate in ordine gerarchico dal generale al particolare. L'ontologia che gli annotatori hanno utilizzato è la seguente: <https://tpas.sketchengine.eu>

Basandoti su quanto ti ho detto, per ciascuna delle seguenti frasi:

- 1. Avrebbe voluto vedere ancora quella luce verde brillare sulla lama e sentirsi tutt'uno con essa.
- 2. Un timido raggio di sole brillò su quei vetri.
- 3. La sua fronte si corruga ed i suoi anelli brillano, poi il bagliore si estende alla sua persona.
- 4. Dopo qualche istante una viva fiamma brillava dinanzi all'apertura.
- 5. Sembra quasi, che qui le stelle brillino di piu'.

Individua il soggetto e/o il complemento oggetto del verbo brillare.

Assegna a ciascuno di essi il tipo semantico più specifico a cui sono riconducibili, utilizzando come riferimento l'ontologia che ti ho fornito e la sua struttura gerarchica.

12.2. Appendix B: Prompt Experiment 2 ChatGPT /Gemini

Sei un linguista che sta lavorando per la risorsa T-PAS, un archivio di Typed Predicate Argument Structures, (anche dette pattern) per l'italiano. Tali pattern sono stati acquisiti manualmente da corpora attraverso l'analisi delle informazioni distribuzionali dei verbi italiani, seguendo la metodologia della Corpus Pattern Analysis descritta in Hanks (2004, 2013). Per ogni verbo presente nella risorsa, T-PAS mostra quali sono i suoi pattern più ricorrenti. I pattern sono costituiti dal verbo e dai suoi argomenti (soggetto, oggetto, complemento preposizionale, ecc.), a ognuno dei quali è assegnata un'etichetta che ne identifica le proprietà semantiche (dette tipo semantico, ad esempio HUMAN, ARTIFACT, EVENT, LOCATION, ecc.). Basandoti su quanto ti ho detto, per ciascuna delle seguenti frasi:

- 1. Partendo dal template scelto e' possibile costruire un sito web con script particolari, grafica aggiuntiva o richieste specifiche.
- 2. Ma come si fa a costruire un sito web che non faccia scappare chi lo visita?
- 3. Ho cominciato da poco a costruire siti web a livello amatoriale soprattutto per passatempo.
- 4. Attraverso lezioni passo per passo e progetti concreti imparerai a costruire siti Web in modo facile e veloce.
- 5. Nello zaino potremo trovare dunque ogni partecipante ha quindi illustrato il contenuto del suo zaino progettando e costruendo un proprio sito web.

Ti chiedo di: 1. Individuare il soggetto e/o il complemento oggetto del verbo costruire. 2. assegnare a ciascuno di essi il tipo semantico più specifico valido per tutte le frasi.

12.3. Appendix C: Example of Gemini output

3	[[HUMAN]] ("Homo Esageratissimus")	[[BEVERAGE]] (lista di cocktail e vodka)
4	[[HUMAN]] ("noi" - sottinteso)	[[BEVERAGE]] ("chardonnay" e "namabiru")
5	[[HUMAN]] ("i monatti")	[[BEVERAGE]] (sottinteso, nel "fiasco")

2. Assegnazione dei Tipi Semantici (Semantic Types)

Per identificare il pattern ricorrente in T-PAS, dobbiamo trovare le etichette semantiche più specifiche che accomunino tutti gli esempi (generalizzazione dai dati):

Soggetto (Arg0): [[HUMAN]]

Sebbene in alcuni contesti letterari o metaforici si possa trovare un animale (es. un gatto che "trinca" il latte), nel corpus analizzato l'agente è sistematicamente un essere umano. In T-PAS, l'etichetta [[HUMAN]] è quella corretta poiché implica la capacità intenzionale di compiere l'azione del bere con l'attitudine tipica di questo verbo.

Oggetto (Arg1): [[BEVERAGE]]

Il tipo semantico più specifico valido per tutte le frasi è [[BEVERAGE]] (bevanda).

- *Perché non [[LIQUID]]?* Sebbene ogni bevanda sia un liquido, il verbo *trincare* seleziona specificamente sostanze destinate al consumo umano (alcolici, birra, vino). Non si "trinca" solitamente l'olio motore o l'acqua piovana, a meno di usi altamente ironici.
- *Nota sulla specificità:* Molti esempi mostrano una preferenza distribuzionale per [[ALCOHOLIC BEVERAGE]], ma per mantenere la validità universale del pattern in un archivio come T-PAS, si preferisce la categoria leggermente più ampia di [[BEVERAGE]], eventualmente annotando la preferenza per gli alcolici nelle note d'uso.