Baltic J. Modern Computing, Vol. 14 (2026), No. 1, 202-216
https://doi.org/10.22364/bjmc.2026.14.1.11

Neural Networks and Knowledge Integration: Ontology-Driven Text Encoding for CLIP

Oleksandr HALUSHKA, Viktor SHYNKARENKO

Department of Computer Information Technology, Ukrainian State University of Science and Technologies, Dnipro, Ukraine

oleksandr.halushka01@gmail.com, shinkarenko_vi@ua.fm

ORCID 0009-0005-3447-2676, ORCID 0000-0001-8738-7225

Abstract. This paper presents a method for improving the quality of multimodal "Text-Image" representations by integrating external ontological knowledge into a text query. The proposed approach integrates semantic and associative knowledge from ontologies directly into the text module of the CLIP neural network model through a tree-like syntax structure, similar to the K-BERT method. Adapting the syntax tree for use as an input parameter of the text module involves modifying the original positional encoding mechanism and attention mechanism. Relevant research was conducted on ways to modify the attention mechanism in accordance with the ontological nature of the syntax tree. The results of the experiments showed that the modified model outperforms the basic CLIP in terms of image classification accuracy in a number of object categories. A particularly noticeable increase in quality is observed for specific classes that require additional conceptual knowledge. The article discusses in detail the model architecture, the process of knowledge enrichment, and the impact of the proposed modifications on the model's performance.

Keywords: ontology, neural networks, deep learning, knowledge graph, knowledge engineering, multimodal representations, software, information technologies.

1. Introduction

Multimodal models trained to semantically match information presented in different modalities have achieved significant success in various combinations of applications, such as "text-audio", "text-image", and "text-video". Existing models show serious results in both downscale and upscale tasks. They can generate video from a text description, or vice versa, summarize the content presented in the video, link text, video, and audio content, forming a single semantic context. This is achieved by projecting information presented in different modalities into a single semantic space formed during the model training process. The organization of this semantic space determines how the model builds associative links between fragments of information.

With some simplification, the process of training such models can be characterized as the process of forming their own implicit ontology based on statistical data. The main role of any ontology, whether explicitly or implicitly defined, is to structurally represent knowledge about a certain subject area so that, using it as a source of knowledge, the

interpretation of specific input information is simplified. Then ontology is essentially a generalized representation of some stable patterns or images formed on the basis of individual concepts with associative connections.

Using ontological knowledge at the level of association building, the idea of using explicitly constructed ontologies (ontologies that were constructed by humans) to enhance the implicit internal ontology of the model makes sense. Then a model that receives not only a text query as input, but also some semantic context, will be able to interpret it more accurately.

However, models trained only on statistical relationships between images and text descriptions may lack knowledge about the subject area and miss some important details. Unlike humans, who can rely on broad contextual experience and facts when recognizing an object, neural network models do not have built-in ontological knowledge or modules and methods for acquiring this knowledge. For example, humans know that a "swan" is a bird that can be white or black, lives in pairs, and is larger than a goose. Such knowledge can help distinguish a swan from other birds that look similar. A neural network model forms such knowledge implicitly, statistically, and exclusively from input data, which may not always be effective, especially for specialized subject areas.

One promising direction for improving the quality and accuracy of models capable of zero-shot classification (Wang et al., 2019) is the integration of knowledge from external structured sources, such as ontologies and knowledge graphs. These approaches allow enriching the information presented in one of the modalities with additional semantic context, which aligns the semantic meaning of the information with the generally accepted understanding of the nature of certain concepts.

Based on this approach, this article presents a modified version of the CLIP model (Radford et al., 2021) (hereinafter K-CLIP) that uses external ontological knowledge to enrich text queries. In K-CLIP, the text encoder is modified so that it can accept as an input parameter a syntactic structure similar to text, but with branches inside, which represents a certain concept enriched with relevant ontological triples. The idea is to allow the CLIP model to use explicit knowledge from external sources to improve the quality of its work.

2. Related Works

Modern deep learning models for matching image and text representations are based on the statistical formation of an understanding of the similarity of information presented in different modalities in the process of training on large volumes of image-text pairs.

One of the well-known basic multimodal models is the CLIP (Radford et al., 2021) (Contrastive Image-Language Pretraining) model proposed by OpenAI. This model is trained to compare two modalities: visual and textual. The model was trained on a large-scale corpus of image-text pairs (about 400 million pairs). In these pairs, the text element describes what is depicted in the image. CLIP consists of two main components – text and visual encoders, each of which constructs a representation of the input information of a certain modality into a high-dimensional vector, which represents the "meaning" of what is depicted in the image or described in the text query. Figure 1 shows a diagram of how the model works.

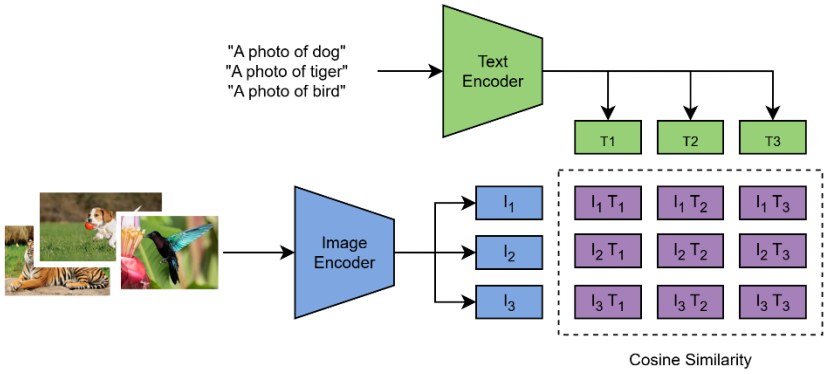

Figure 1. CLIP model workflow

For things that are similar in meaning in the text and in the image, CLIP will construct similar vector representations. Cosine similarity is used as a similarity metric, which compares the direction of vectors relative to each other:

$$Similarity = cos(\theta) = \frac{R_T \cdot R_I}{||R_T||\,||R_I||}, \quad (1)$$

where R_T is the representation of the text query, R_I is the representation of the image, $||R_T||$ and $||R_I||$ are norms of the vectors of the text and visual representations, respectively. The cosine similarity values determine the following properties:

- values close to one indicate semantic correlation between concepts;
- zero values reflect the orthogonality of vectors in multidimensional space, which is interpreted as the semantic independence of concepts;
- negative values close to -1 indicate semantic opposition between concepts.

Using this model, classification can be performed based on similarity metrics. Interestingly, since the model built an implicit ontology during training, which it uses to understand the world, it is capable of zero-shot classification (classification of previously unknown classes that were not represented in the training sample) without additional training.

With its release, CLIP became a landmark work in the field of multimodal representations. The model demonstrated that training on unlabeled correspondences between two modalities can result in universal representations that are independent of a specific modality. A similar approach, ALIGN (Jia et al., 2021), was proposed by Google, where an even larger corpus of image pairs and their text descriptions was used for training. This also led to improved representation quality, which in turn improved results on tasks such as image classification and search.

However, neither CLIP nor ALIGN explicitly use external ontological knowledge. Their internal understanding of the world is formed statistically and therefore does not guarantee the assimilation of specific facts or conceptual hierarchies of concepts.

Early work on training recognition models with semantics in mind showed the advantages of using structured, explicitly specified knowledge. For example, the DeViSE model (Frome et al., 2013) mapped images and words to a common semantic space using

vector representations of words, which allowed objects to be classified even if they had not been encountered during training. This was achieved through the semantic proximity of their names. Another approach is to use a knowledge graph to improve classification (Marino et al., 2016). This approach is based on the use of a graph neural network whose configuration is built on the basis of the topology of a knowledge graph or ontology. This allows the use of knowledge about the relationships between objects in the process of image recognition, which has proven beneficial in multi-class classification.

In the field of natural language processing, the integration of external knowledge has been studied for a long time. The BERT model, which was trained on a large-scale text corpus, does not contain explicit facts. To solve this problem, some modifications of this model were created, such as ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019), and K-BERT (Liu et al., 2020). According to the results, these modified models show better performance on tasks that require specialized knowledge compared to the basic BERT model.

For example, KnowBERT embeds knowledge from WordNet (Fellbaum, 2010) directly into the model parameters, which obviously requires additional training, complicating integration. K-BERT, on the other hand, uses external ontological knowledge as context. K-BERT is most similar to K-CLIP. It differs from other approaches in that it does not require training adapters that translate ontological triples into vector representations. Instead, it enriches the input text with additional ontological knowledge. This process consists of three main stages: recognizing named entities in the text, searching for ontological facts in the knowledge base that correspond to the recognized entities, and embedding the found facts into the sentence structure of the text. Thus, in general, the text retains its basic structure and acts as a kind of skeleton onto which additional information in the form of knowledge is built. However, despite the conceptual similarity in using visibility matrices for knowledge integration, K-BERT and K-CLIP operate in fundamentally different settings: K-BERT modifies BERT, a text-only encoder designed for NLP tasks, and requires task-specific fine-tuning, whereas K-CLIP modifies CLIP, a vision-language model, and operates without additional training. This architectural difference makes direct quantitative comparison between the two approaches not meaningful.

There is also another approach that modifies the CLIP model through prompt tuning using knowledge. The idea is to use the generation of text prompts that describe a certain class of objects based on explicitly specified knowledge (Kan et al., 2023).

A body of previous work points to the potential effectiveness of applying ontological knowledge to improve the performance of multimodal models. The K-CLIP modification continues this work in this direction, architecturally integrating the use of ontological knowledge into the CLIP model for image classification tasks.

3. Methodology

This section describes the architecture of the modification of the original CLIP model to enable the integration of external ontological knowledge. CLIP consists of two encoders: visual (for images) and text (for captions or class names). We leave the visual encoder unchanged, while focusing our modifications on the text encoder.

3.1. Model structure and text enrichment using knowledge

The CLIP text encoder belongs to the family of encoder-only transformers (Singh, 2024). Such transformers accept some data as input and generate their representation in the form of contextually enriched high-dimensional vectors as output. Similar to the BERT model (Devlin et al., 2019), the semantic representation of the entire sequence of tokens as a whole, rather than each token individually, is stored in a special reserved token <CLS>, which is used as a vector representation of the input text. In K-CLIP, the input to the text encoder is not only the concept, but also the tokens added to it that represent knowledge.

The process of knowledge enrichment occurs as follows:

- the knowledge base is searched for ontological facts in which the concept appears as a subject or object;
- the facts found are selected taking into account the degree of semantic depth and quantity;
- a tree-like syntax structure is formed, in which the initial concept with branches acts as the root.

Fig. 2 shows an example of a tree-like syntactic structure built on the basis of an ontological knowledge segment.

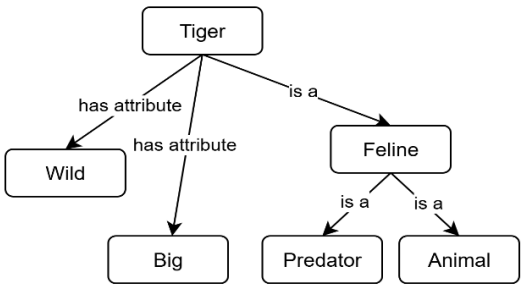

Figure 2a. Ontological segment

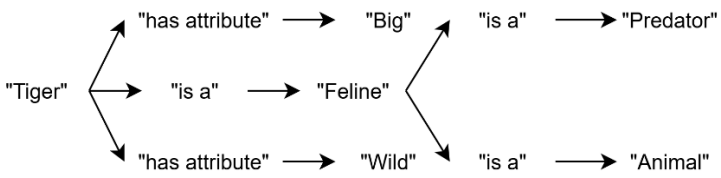

Fig 2b. Tree-like syntax structure based on the segment shown in Figure 2.a

The nature of the knowledge used should, to one degree or another, determine the attributive, semantic, and associative context of the concept.

The structure of the constructed syntactic structure does not coincide with the structure of ordinary text written in natural language. In ordinary texts, each word has its own unique position in the sequence, based on which it can be uniquely identified.

The tree-like structure requires the introduction of relative positioning of tokens in the sequence according to the tree topology. The need for relative positioning arises from the structural difference between linear text and tree-based ontological representations. In natural language, word order is inherently sequential and semantically significant. In contrast, a tree constructed from ontological triples exhibits hierarchical rather than sequential semantics: the order within each triple matters, but different branches representing independent facts have no inherent ordering. Applying standard positional encoding to a flattened tree would incorrectly suggest sequential relationships between topologically independent nodes, distorting the semantic structure that the ontology encodes.

The point is that when transforming a tree into a sequence, each token must be assigned two indexes: actual and relative. The actual index is the position of the token in the entire sequence, which does not affect anything due to the parallel nature of the transformer architecture - it does not matter in what order we submit tokens to the input, but how the positional encoding of tokens is performed. The relative index is the position of the token relative to the initial token, which is determined topologically based on a tree-like structure. The CLIP text encoder uses trained embedding as positional encoding, i.e., by default, the token representation t with index i is appended with positional embedding p with the same index, where i is the actual position of the token in the input sequence.

$$e[i] = x[i] + p[i], \tag{2}$$

where $e[i]$ is the resulting vector of element-wise addition of the components of the token representation x and the corresponding positional embedding p.

Adaptation of the syntactic structure with built-in additional knowledge involves modifying the positional encoding by selecting the corresponding positional embedding using its relative index instead of the actual one. Fig. 3 illustrates both the original CLIP positional encoding mechanism and the proposed modification for tree-structured input.

This approach allows us to approximate the representation of related concepts and the relationships between them at the level of perception by the trained transformer model. Transforming a tree-like structure into a simple sequence destroys the semantic structure, so positional modification helps the model to correctly perceive information without violating its semantic coherence. As discussed above, standard positional encoding assigns sequential positions that do not reflect the tree topology. Consequently, tokens from independent branches appear sequentially adjacent, causing the model to establish spurious semantic connections between unrelated concepts.

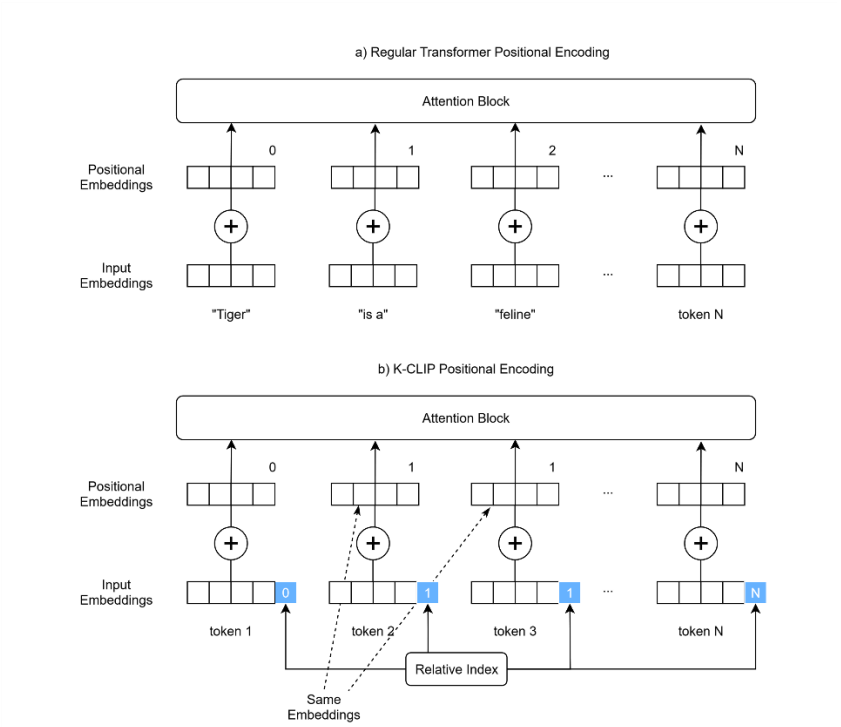

Figure 3. Positional encoding: (a) original CLIP mechanism using actual token positions; (b) K-CLIP modification using relative positions based on tree topology

However, this implementation raises the following problem: different nodes of the tree-like structure with the same relative positional indices will be perceived by the model as being next to each other, which leads to variability in token following. This problem is known as “knowledge noise” and can be described as a situation where the model, using the attention mechanism, establishes a connection between tree nodes that are located in different branches and should not interact with each other in any way. To solve this problem, as well as to control the influence of knowledge, a modification of the attention mechanism is introduced.

3.2. Attention distribution control

The attention mechanism as a concept is a key component of any model built on the transformer architecture. It allows you to form vector representations of tokens in a sequence based on their own context. This means that each token in the sequence correlates to some extent with every other token in that sequence, i.e., the representation of each token distributes its attention among the other tokens in the sequence, including itself. The degree of this distribution allows us to calculate the contribution of each token to the resulting contextual representation of that token.

In the context of K-CLIP, the modification of the attention mechanism consists in limiting the distribution of attention to those tokens that are topologically located in

different branches of the syntactic structure. Each token should only see those tokens that are in the same branch as it is relative to the main concept. The limitation of attention is that topologically distant tokens do not affect the calculation of each other's contextual representation, which prevents the distortion of the semantic meaning of the entire information as a whole. The attention distribution control scheme is illustrated in Fig. 4.

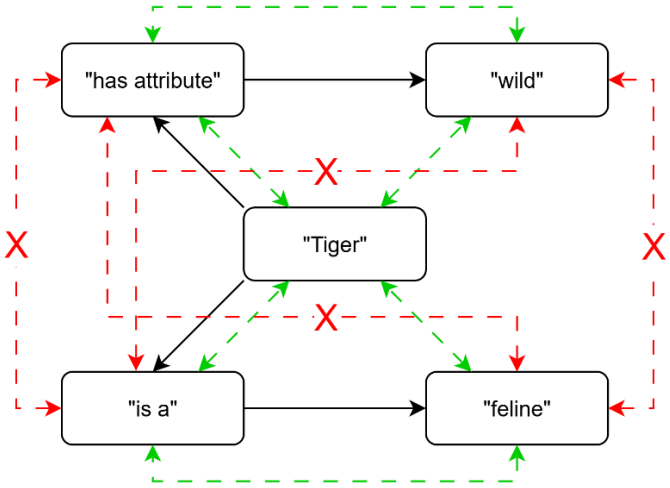

Figure 4. Example of attention distribution control

The formal implementation of the modification was done by constructing a binary visibility matrix VM, in which each element $VM_{i,j} = 1$, if the tokens i and j are in the same branch, and $VM_{i,j} = 0$ if they are in different branches. In terms of size, this matrix coincides with the attention matrix, so to put it into action, each element of the attention matrix A must be multiplied by the corresponding element of the matrix VM:

$$MA = A \circ VM, \tag{3}$$

where MA is the modified attention matrix formed by element-by-element multiplication of the corresponding elements of the matrices (operation $\circ$), A is the original attention matrix, and VM is the visibility matrix.

The visibility matrix dimensions are bounded by the CLIP text encoder's context window of 77 tokens, resulting in a maximum matrix size of 77 × 77. After accounting for special tokens (<CLS>, <SEP>) and the root concept, approximately 72 tokens remain available for ontological knowledge. Given that a typical ontological triple requires 6–10 tokens after tokenization (subject, predicate, and object with potential subword splitting), the practical ontology size is constrained to 7–12 triples per query. In our experiments, we used 10–12 facts per concept, which approaches this architectural limit while maintaining representational quality. This constraint does not fundamentally limit the number of relationships that can be encoded, but rather requires careful selection of the most

semantically relevant triples for each concept.The algorithm for constructing the visibility matrix is presented below:

Algorithm 1. Visibility Matrix Construction

```
Input: tree T with root r, token sequence {t₁, ..., tₙ}
Output: visibility matrix VM ∈ {0,1}^(nxn)

1: for each token tᵢ do
2:   path[i] ← ancestors(tᵢ, r) ∪ {tᵢ}  // path from root to tᵢ, including tᵢ
3: end for
4: for i = 1 to n do
5:   for j = 1 to n do
6:     if tᵢ ∈ path[j] OR tⱼ ∈ path[i] then
7:       VM[i,j] ← 1  // tokens are in the same branch
8:     else
9:       VM[i,j] ← 0  // tokens in different branches
10:    end if
11:  end for
12: end for
13: return VM
```

An example of a generated visibility matrix is shown in Fig. 5.

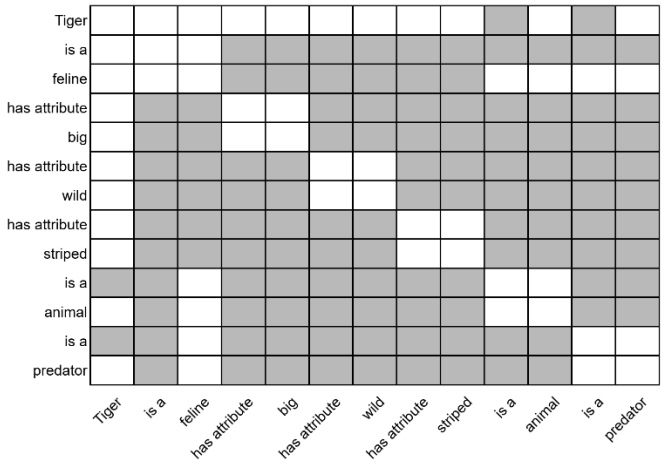

Figure 5. Visibility matrix

This approach to controlling the distribution and concentration of attention ensures that the main root concept will play a decisive role in the resulting representation, and that the additional knowledge with which this concept has been enriched will not interact with each other at the level of attention. This follows from the very idea of enriching some input data with external knowledge. Knowledge should semantically “guide” the representation of the concept closer to its semantic meaning, rather than obscuring the initial representation.

4. Experiments

To evaluate the effectiveness of the approach, a series of experiments was conducted on image classification tasks for different categories and classes of objects. The CLIP model with a visual encoder based on ViT-B/32 (Awofeso, 2024) and a text encoder with a vector dimension of 512 (the original model) was used as a basis. The model was initialized with the trained parameters of the CLIP model without additional training. Small knowledge graphs with approximately 10-12 facts each were used as ontological knowledge. The triplets of these graphs describe primarily visual attributes, as well as hierarchical and semantic components of the concept.

The experiments were conducted in three stages:

- testing the variability of the attention mechanism modification for different blocks in order to determine how much the modification at different levels of abstraction affects the quality of the model's performance;
- testing the model for the effectiveness of object classification using additional knowledge as a source of additional clarification and awareness in order to establish the effectiveness of the influence of external knowledge on the quality of classification;
- testing the model's ability to perform fine-grained discrimination between visually similar object classes (different dog breeds) by comparing three encoding strategies: generic concept baseline, K-CLIP with breed-specific ontological attributes, and direct breed naming.

The first stage is to check how modifying attention on different sequential blocks of the Multi Head Attention transformer component affects the result. The CLIP model, like any other model based on transformer architecture, includes a Multi Head Attention component, which is represented as a sequence of interconnected attention blocks. A typical model usually has 12 blocks. The sequential connection of blocks allows attention to be formed at different levels of abstraction of information perception. The first blocks establish connections between atomic fragments, while the latter direct attention between more abstract concepts represented in the input token sequence. Tests were conducted on different configurations in which some blocks were modified while others were not.

In this experiment, we investigate the effect of applying our modifications at different layers of the transformer. The K-CLIP approach involves two components: (1) relative positional encoding, which is applied uniformly to all input tokens, and (2) attention masking via the visibility matrix, which can be selectively enabled at specific transformer blocks. In these experiments, we vary which blocks receive the attention masking while keeping relative positional encoding constant. The tested configurations are summarized in Table 1.

Table 1. Attention masking configurations

Configuration	Modified Blocks	Description
First only	Block 1	Masking at the lowest abstraction level
Middle only	Block N/2	Masking at intermediate abstraction level
Last only	Block N	Masking at the highest abstraction level
Both ends	Blocks 1 and N	Combined low and high level masking
All blocks	Blocks 1-N	Full masking throughout all layers
None	-	Baseline

Such variability in configurations helps to reveal the dependence of the effect of attention modification on the level of abstraction of representation. This provides an understanding of in which cases and under which configurations attention modification in “manual mode” can take place, and how this affects the quality of the model's performance. The main dataset was used as a test data set, the data of which was divided into three categories: animals, rare birds, and rare things. The animals category included the following object classes: cat, dog, eagle, elephant, ostrich, rabbit, tiger; the rare birds category included the following object classes: helmeted hornbill, imperial woodpecker, kakapo, purple-throated carib, california condor; the rare things category included the following object classes: bathynomus giganteus, grandidierite, hallucigenia sparsa, welwitschia mirabilis. For the experiments, 10 images were selected from each class (160 in total). Fig. 6 shows some examples of images from the test data set.

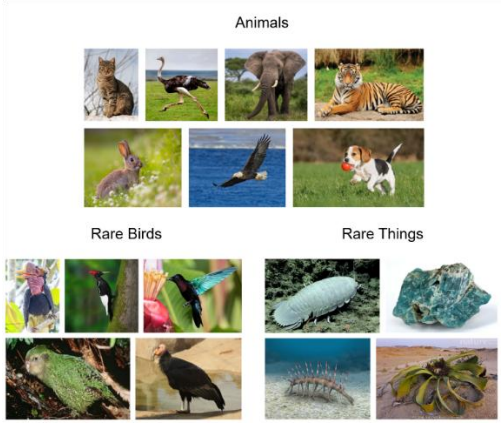

Figure 6. Object categories

The dataset from the previous stage was used for the second stage of the experiments. The quality of classification was evaluated by the relative deviation of the cosine similarity value of the modified model from the value obtained from the original model. Given that cosine similarity is used, a positive deviation indicates an improvement in quality, and a negative deviation indicates the opposite, since when comparing information of different modalities with the same semantic meaning, we expect the ideal result to be close to unity.

During the third stage, the model was tested for its ability to distinguish between different dog breeds. Five dog breeds were selected: German Shepherd, Pekingese, Saint Bernard, Welsh Corgi, and Yorkshire Terrier. Ontologies were formed for each breed, each of which includes 10-12 facts that focus more on the original visual attributes characteristic of a particular breed. Twenty images of dogs were also selected for each breed (100 in total).

The quality of how well the model distinguishes one breed from another was assessed by comparing cosine similarity values across three encoding strategies: using only the generic base concept "dog" (same for all breeds), using K-CLIP with breed-specific ontological knowledge, and using the direct breed name. This experimental design allows evaluating the contribution of ontological knowledge to fine-grained discrimination.

5. Results

5.1. Variative modification of attention

The results of the first stage of experiments show that the trained model significantly degrades in quality when modifying the initial blocks responsible for more specific and atomic aspects of the input information. Modification of the first block in all cases and combinations gives a negative result of approximately -47% relative to the original model without the use of external knowledge. At the same time, modifying attention only on the last block gives a positive effect and an increase in the quality of the model's performance by 3.16% relative to the original model. Fig. 7 shows the results of the modified model with different configurations of block modifications. The indicator is the increase in representation quality relative to the original CLIP model. Negative values indicate a deterioration in performance, while positive values indicate an improvement in the quality of representation construction.

The results show that manipulating the attention mechanism is very sensitive and can only be applied to the final blocks when it is necessary to slightly direct attention in the right direction without destroying the basic structure of the attention calculation process and the connections between the parameters of the trained model. The sensitivity stems from how pre-trained transformers organize information processing. Early attention blocks have learned to capture local syntactic patterns and basic token dependencies; disrupting these patterns invalidates the representations that all subsequent layers expect as input. Final blocks, however, refine already-coherent semantic representations, making them more tolerant to controlled modifications that guide attention toward specific semantic aspects without breaking the model's internal consistency. Importantly, this approach allows engineering semantic relationships without requiring explicit knowledge of which internal vectors correspond to which concepts. Instead of manipulating representations directly, we control relationships structurally through the visibility matrix, leveraging the transformer's own attention mechanism to respect ontological structure.

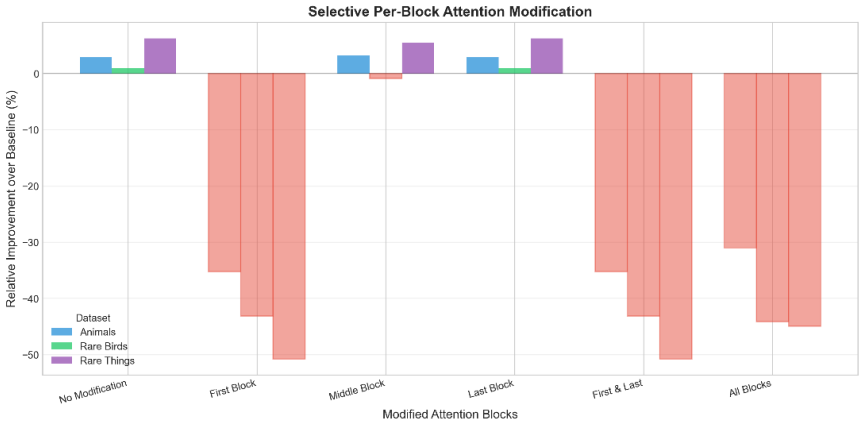

Figure 7. Modification of attention in different configurations

5.2. Classification

Tests of the model's ability to classify and compare its results with the initial unmodified model show a positive increase in efficiency, especially for those classes of objects that are rarely found in the information space and have names that are not prone to strong associative reflections. Thus, the rare things category showed an average increase of 6.22%, which is the best indicator compared to the other two categories, animals and rare birds, with results of 2.5% and 1.3%, respectively. It should be noted that generating text descriptions based on ontology using a third-party LLM gives better results than embedding knowledge. The average increase in efficiency compared to the unmodified model for the categories animals and rare things was 6.5% and 6.75%, respectively. These results can be explained by the fact that the original model was trained specifically on text descriptions, so it perceives plain text better than adapted ontological triples. The results of the experiment are presented in Table 2.

Table 2. Classification of different categories of objects

Category	S_1	S_2	S_3	$S_2 \rightarrow S_1$, %	$S_3 \rightarrow S_1$, %
Animals	0.28	0.29	0.30	2.50	6.50
Rare birds	0.33	0.33	0.33	1.30	0.78
Rare things	0.30	0.32	0.32	6.22	6.75
Average	0.30	0.31	0.31	3.15	4.88

In the table: S_1 is cosine similarity between the image and the object name calculated by the original model, S_2 is the modified model using knowledge embedding, S_3 is the similarity between the image and the synthetically generated description based on ontological triples.

5.3. Discrimination of subtle object categories

Assessing the model's ability to distinguish between conceptually and semantically similar classes of objects demonstrates the value of additional knowledge in fine-grained discrimination tasks. We compare three encoding strategies: S_1 uses only the generic concept "dog" as a baseline, S_2 uses K-CLIP with breed-specific ontological attributes, and S_3 uses the direct breed name. This progression allows us to isolate the contribution of structured ontological knowledge ($S_2 \rightarrow S_1$) from the upper bound achievable with explicit naming ($S_3 \rightarrow S_2$). The results are presented in Table 3.

Table 3. Distinguishing between similar classes of objects

Breed	S_1	S_2	S_3	$S_2 \rightarrow S_1$, %	$S_3 \rightarrow S_2$, %
German Shepherd	0.27	0.30	0.31	+12.1	+5.8
Pekingese	0.26	0.30	0.35	+13.4	+15.4
Saint Bernard	0.26	0.27	0.33	+4.8	+20.1
Welsh Corgi	0.27	0.29	0.32	+6.7	+11.2
Yorkshire Terrier	0.25	0.28	0.34	+10.8	+19.3
Total	0.26	0.29	0.33	+9.6	+14.4

In the table S_1 is the cosine similarity using only the base concept "dog" (same embedding for all breeds), S_2 is the similarity using K-CLIP with breed-specific ontological attributes, S_3 is the similarity using the direct breed name. The columns $S_2 \rightarrow S_1$ and $S_3 \rightarrow S_2$ show the relative improvement in similarity.

6. Conclusions

This paper presented the K-CLIP model, a model that uses external ontological knowledge as a source for semantic enrichment of input text information. The approach is based on the integration of ontological triples into a text encoder by introducing a mechanism for topological positioning of tokens and controlling the distribution of attention flow in a trained model built on the basis of transformer architecture. This solution allows the model to take into account semantic relationships based on the ontological facts provided to it when comparing images with textual information, which brings its behavior closer to the human way of comparing multimodal information: when a person sees an object, they also use their previous experience as a source of knowledge, which helps them identify the object more accurately and confidently.

The experiments confirmed the practical promise of this approach, as the modified model outperformed the original in a series of classification tests, especially where it was necessary to distinguish subtle semantic differences, such as in the experiment on classifying different breeds of dogs, where in each syntactic tree structure the concept of "dog" served as the initial concept, and the differences were only in the additional knowledge that was attached to this concept depending on the specific breed.

The study opens up opportunities for further work in the direction of using formal knowledge as a way to improve the efficiency of neural network models. In the future, it is planned to improve methods of using structured knowledge to increase the interpretability of deep learning models and their transition from simple pattern recognition to semantic understanding of context.

References

Awofeso, Z. (2024). An explanation of the Vision Transformer (ViT) paper. Medium. https://medium.com/codex/an-explanation-of-the-vision-transformer-vit-paper-8cdd399741aa

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, volume 1 (long and short papers) (pp. 4171-4186).

Fellbaum, C. (2010). WordNet. In Theory and applications of ontology: computer applications (pp. 231-243). Dordrecht: Springer Netherlands.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26.

Jia, C., Yang, Y., Xia, Y., Chen, Y. T., Parekh, Z., Pham, H., ... Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning* (pp. 4904-4916). PMLR.

Kan, B., Wang, T., Lu, W., Zhen, X., Guan, W., Zheng, F. (2023). Knowledge-aware prompt tuning for generalizable vision-language models.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P. (2020). K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 03, pp. 2901-2908).

Marino, K., Salakhutdinov, R., Gupta, A. (2016). The more you know: Using knowledge graphs for image classification. arXiv preprint arXiv:1612.04844.

Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., Smith, N. A. (2019). Knowledge enhanced contextual word representations. arXiv preprint arXiv:1909.04164.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

Singh, R. (2024). Types of Transformer Model. Medium. https://medium.com/@RobuRishabh/types-of-transformer-model-1b52381fa719

Wang, W., Zheng, V. W., Yu, H., Miao, C. (2019). A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology* (TIST), 10(2), 1-37.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129.

Received October 27, 2025, revised December 18, 2025, accepted February 20, 2026