



HAL
open science

Web Image Context Extraction with Graph Neural Networks and Sentence Embeddings on the DOM tree

Chen Dang, Hicham Randrianarivo, Raphaël Fournier-S’Niehotta, Nicolas Audebert

► **To cite this version:**

Chen Dang, Hicham Randrianarivo, Raphaël Fournier-S’Niehotta, Nicolas Audebert. Web Image Context Extraction with Graph Neural Networks and Sentence Embeddings on the DOM tree. GEM: Graph Embedding and Mining - ECML/PKDD Workshops, Sep 2021, Bilbao, Spain. pp.258-267, 10.1007/978-3-030-93736-2_20. hal-03324009

HAL Id: hal-03324009

<https://hal.science/hal-03324009>

Submitted on 25 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Web Image Context Extraction with Graph Neural Networks and Sentence Embeddings on the DOM tree

Chen Dang¹, Hicham Randrianarivo¹,
Raphaël Fournier-S'niehotta², and Nicolas Audebert²

¹ Qwant SAS, Paris, France

chdangg@gmail.com, h.randrianarivo@qwant.com

² CEDRIC (EA4629), CNAM Paris, HESAM Université, France
fournier@cnam.fr, nicolas.audebert@cnam.fr

Abstract. Web Image Context Extraction (WICE) consists in obtaining the textual information describing an image using the content of the surrounding webpage. A common preprocessing step before performing WICE is to render the content of the webpage. When done at a large scale (*e.g.*, for search engine indexation), it may become very computationally costly (up to several seconds per page). To avoid this cost, we introduce a novel WICE approach that combines Graph Neural Networks (GNNs) and Natural Language Processing models. Our method relies on a graph model containing both node types and text as features. The model is fed through several blocks of GNNs to extract the textual context. Since no labeled WICE dataset with ground truth exists, we train and evaluate the GNNs on a proxy task that consists in finding the semantically closest text to the image caption. We then interpret importance weights to find the most relevant text nodes and define them as the image context. Thanks to GNNs, our model is able to encode both structural and semantic information from the webpage. We show that our approach gives promising results to help address the large-scale WICE problem using only HTML data.

Keywords: Web Image Context Extraction · Information Retrieval · Graph Neural Network · Natural Language Processing

1 Introduction

Searching for images on the Web is essential for Internet users. Then, there is a need for efficient indexing methods to process large quantities of images. A useful step for the indexation in an image retrieval system consists in identifying the part of a webpage's text that best describes the image. The problem of extracting this *context* from the webpage is called Web Image Context Extraction (WICE) cf. Fig. 1. Visually rendering the webpage facilitates the extraction of an image's context, by loading and placing all structural elements of the page at the cost of evaluating several scripts. On a large scale, visual rendering and content

extraction from a webpage is not tractable. We investigate how the HTML data structure may help in extracting images’ contexts.



Fig. 1. An example of the WICE setting: an image, and its textual description (*context*) in the webpage.

Many approaches to WICE have been proposed. [17] use metadata related to the image as the textual context. Defining the context as the text in a “window surrounding the image” in the HTML is common, and some works try to find an optimal number of words to extract around the image [2,5]. [6] consider multiple sources of text, *e.g.*, title and meta information, as the context. These text-based methods often result in incomplete sentences, and do not provide accurate context when the context and image are not close in the HTML file.

Structure-based approaches focus more on the structure of the HTML document. An HTML document can be describe as a tree structure where each tag or text an object and the nested objects are “children” of the enclosing one wich is called the Document Object Model (DOM) [21]. Relying on the DOM tree, some works [7,8] measure similarities between the alternative text of the image and other texts, or develop precise webpage segmentation rules. [12] propose a *broadcast model* which combines the text blocks around images and information from other webpages linked to the webpage. [4] classify webpage’s structures into three categories and handcraft rules to extract context. With a high focus on the page structure, DOM-tree based approaches often ignore or fail to fully use textual content, and recent evolutions of webpage programming have rendered many methods based on hard-coded rules no longer applicable.

Finally, some approaches focus on the webpage’s visual layout. [1] use visual information to perform the segmentation using a set of predefined rules. [19] propose to extract all text which includes a caption and an alternative text at the same level as the image in the DOM tree. They also keep the texts around the image within a radius of 0.3 of the webpage rendering height. [18] compare visual and semantic clustering and find that visual based clustering performs much better in extracting information of web images

Besides, many graph-based information retrieval methods have been recently proposed. [14] use a graph-based framework to capture non-local and non-sequential context in sets of sentences. [3] introduce a Graph Neural Network model for multi-step reasoning. [13] study relation extraction for semi-structured websites. [18] compares visual and semantic clustering and find that visual based clustering performs much better in extracting information of web images.

However, browser-based rendering is costly due to CSS and Javascript processing. It is nonetheless required by the state-of-the-art visual WICE approaches

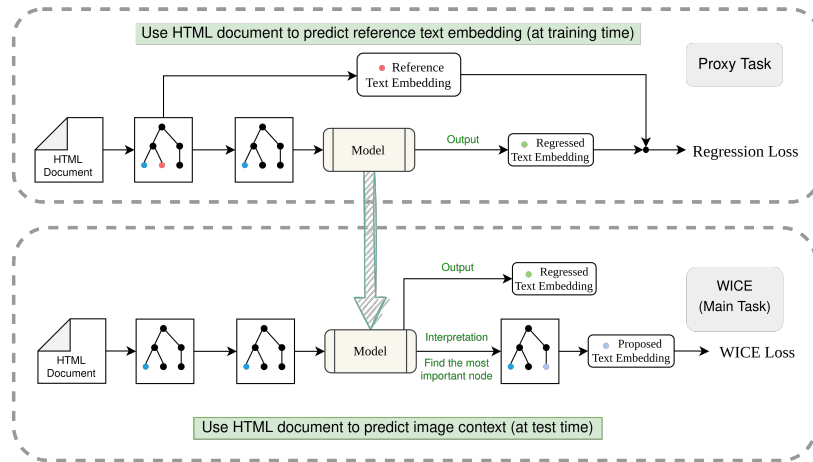


Fig. 2. Our pipeline. A graph neural network is first trained in the proxy task to predict the input HTML document’s reference text (red dot), and then the importance score of the nodes (blue dots) in the prediction process is computed by interpreting the GCN model during the main task. The most predominant textual node (green dot) is then used as the context of the image.

that rely on the full styled layout. We model webpages as graphs whose features are node types and text content. More specifically, our contributions to the problem of large-scale WICE are:

- Inject semantics into the DOM tree using state-of-the-art language models to generate sentence embeddings for each text node,
- Model webpages as graphs and use sentence embeddings as node features to train a graph neural network combining structural and semantic information,
- Use graphical models for large-scale processing of highly diverse news websites.

2 Method

Our goal is to identify nodes in the DOM tree that may contain part of an image’s context. Since the DOM tree is a graph, we may use graph convolution networks (GCNs) [9] to do so. However, to bypass the lack of labeled datasets, we propose a *proxy task*, on unlabeled HTML documents to train our model. The results of this task may be interpreted to solve the WICE problem. Fig. 2 illustrates the sequence of steps in our method.

For each HTML document that contains an image, we use the longest text between the alternative text ³, the caption ⁴, and the image title ⁵ as the

³ "alt" attribute of the `` tag, which provides descriptive information of the image.

⁴ text in `<figcaption>` that usually displays a short explanation besides the image.

⁵ "title" attribute of the `` tag

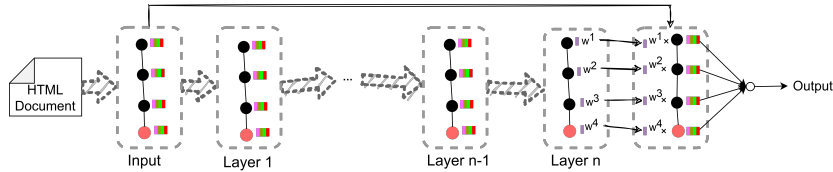


Fig. 3. An illustration of the wGCN (weight-GCN) architecture. The nodes in the last layer have a feature dimension of 1. We interpret it as the importance score of each node. The output is the weighted average of the text embeddings of all nodes.

reference text. We assume here that the reference text always describes the image. This assumption may potentially cause bias.

While crawling news websites, we empirically found that approximately 50% of them provide a caption or a reference text for their main illustrative images. This allows us to train on a relatively large corpus and also underlines the need for WICE in the wild: many websites do not provide clear textual contexts for their images. Training models on a proxy task instead of directly predicting the context has many advantages. Firstly, we may rely on unlabeled datasets to train the model and solve the WICE problem, avoiding the need for annotations. It is easier and cheaper to crawl webpages with captions than manually annotate context sentences in thousands of articles. Moreover, at test time, our model will be used to infer the missing reference text: this means we can perform WICE even when the reference text is not present, which is where it is the most needed.

2.1 Framework

To extract the textual context of an image, we want to assign a weight w_i to all text nodes n_i from the DOM graph \mathcal{G} . Since we cannot learn w_i directly, we use a proxy task that consists of regressing a global embedding $\hat{\mathbf{z}}$ for the whole page. We use as a supervised target the sentence embedding \mathbf{z}^* of the reference text (alt-text or caption) extracted for the image.

For most of our models, we will assume that it is possible to reconstruct the reference text’s embedding using a linear combination of the other text embeddings, *i.e.*, we consider outputs in the form $\hat{\mathbf{z}} = \sum_i w_i f(\mathbf{z}_i)$ where \mathbf{z}_i is the embedding for the i -th node. Since the regressed text embedding is obtained by averaging weighted node embeddings, we assume that the largest contributor, *i.e.* the text node with maximum weight, is the most relevant for indexing. The image context can thus be obtained by taking the text node n_{target} that verifies $n_{\text{target}} = \arg \max_i w_i$

In this work, the context is the most important text node, although this notion could be extended to all nodes over some threshold.

As a WICE metric for the extracted context, we compute the cosine similarity between the chosen text node’s sentence embedding with the reference text’s embedding, which is commonly used to measure document similarity in NLP.

2.2 Baselines

Inspired by the literature on WICE, we define various baselines for both the proxy and the WICE tasks. We define the **distance** baseline, which weights a text node by the inverse of its distance d_i to the image node in the graph:

$$\hat{\mathbf{z}} = \sum_i w_i \mathbf{z}_i = \sum_i \frac{1}{d_i} \cdot \mathbf{z}_i \quad (1)$$

Implicitly, this baseline also defines an equivalent WICE baseline, which we call **text after image**, where the textual context of the image is the closest text node in the graph. This is known as "window-based context extraction" in the WICE literature [2]. As a simpler rule-based baseline, we also consider the **title** WICE heuristic that uses the **title** attribute of the webpage as context.

We define a **blind** WICE baseline as selecting the node with the most similar text (in the embedding space) across all text nodes, compared to the embedding regressed by the network ($w_i = 1$ if $i = \arg \max_j (\cos(\hat{\mathbf{z}}, \mathbf{z}_j))$ and 0 otherwise). Intuitively, it can be interpreted as "looking for the missing caption". In this baseline, the WICE is "blind" since the reference text is completely unseen. This defines a lower-bound for our method: as a sanity check, the most important node we find should at least describe the image better than the predicted embedding.

Finally, we define an **oracle** baseline that uses the actual reference text to find the most similar node. The resulting cosine similarity gives us an upper bound of what is achievable using our proxy model. It may be used as an indicator representing the model's performance potential: the smaller the gap between a model's results and the **oracle**, the better its performance.

2.3 Models

Two base models are studied to perform the text regression: Graph Convolution Network (GCN) [9] and Graph Attention Network (GAT) [20]. GCNs are well-known for their promising performance on graph data. However, it is difficult to explain the prediction because the fusion of graph structure and feature dimension achieved by GCN is an explicitly irreversible process [22]. In comparison, GAT is a well-performing model, interpretable thanks to its attention mechanism which can be used as the weight of the text block. Its multi-headed mechanism can also be utilized to stabilize the performance of the model.

We study two different approaches to help interpret the GCNs and produce the node weights vector \mathbf{w} . First, we propose a GCN model that explicitly assigns weights to the nodes to facilitate the model's explanation, referred to as weight-GCN (wGCN). A traditional GCN would map the entire graph \mathcal{G} to the target embedding: $\hat{\mathbf{z}} = \Psi_\theta(\mathcal{G})$ where Ψ is the GCN with parameters θ . However the information of which nodes contributed the most to the predictions is lost. Instead, we make the GCN produce one weight per node. The regression result is then the weighted average embedding of all nodes in the graph. Fig. 3 illustrates the principle of the wGCN. Formally, let Ψ_θ be the GCN, $\mathbf{w} = \Psi_\theta(\mathcal{G})$ the output

vector of node weights, and \mathbf{z}_i be the text embedding of the node i of graph \mathcal{G} . Then the regressed text embedding $\hat{\mathbf{z}}$ can be denoted as:

$$\hat{\mathbf{z}} = \sum_i \Psi_\theta(\mathcal{G})[i] \cdot \mathbf{z}_i = \sum_i w_i \mathbf{z}_i \quad (2)$$

The wGCN is then trained using backpropagation and stochastic gradient descent: $\Psi_\theta^* = \arg \min_\theta \mathcal{L}_\Psi(\theta)$. \mathcal{L} is the proxy task loss function. We minimize the negative cosine similarity between $\hat{\mathbf{z}}$ and the reference text \mathbf{z}^* *i.e.*:

$$\mathcal{L}_\Psi(\hat{\mathbf{z}}, \mathbf{z}^*) = 1 - \cos(\hat{\mathbf{z}}, \mathbf{z}^*) = 1 - \cos\left(\sum_i w_i \mathbf{z}_i, \mathbf{z}^*\right) \quad (3)$$

Our second approach uses the GAT attention scores as the weights \mathbf{w} . The key difference between the GAT and wGCN is that wGCN learns the relationships between nodes and produces one weight per text node. The regression embedding is then an average of the embeddings weighted by the wGCN scores. In comparison, attention scores are only indirectly linked to the output embedding.

We also used the DeeperGCN (DGCN) architecture [10,11] to create deeper GCN models. DGCN aims at solving common problems affecting deep GCNs, such as vanishing gradients, over-smoothing, and overfitting issues. DGCN also uses recent deep learning tricks such as residual learning or dilated aggregation.

In our study, deeper neural networks make more sense because more neighbors can be explored. Deeper networks lead to more nodes visited by the central node, and therefore more information collected. This way, the image node receive information from all the article’s nodes, even for pages with complicated DOM structures; thus improving the representation capacities of the model.

3 Experiments and results

The dataset for our study was constructed using webpages from the Qwant News search index⁶. It consists of 242 247 webpages from 1341 different websites, crawled mainly from French news websites. Some Italian, German, and Spanish websites are also included. Both international and regional websites from France are included in order to maximize diversity.

We preprocess the HTML documents as follows: the content in `<main>`, `<body>` or `<article>` is first extracted from the webpage. Some tags pertaining to layout, such as `<style>` or `<button>`, are then removed to clean up the DOM tree from unnecessary nodes. Then, we extract the biggest image (in pixels) of a webpage with its reference text. Webpages without such an image are removed from the dataset. After preprocessing, the datasets contains 119 550 webpages in 805 websites. Texts are encoded using the multilingual sentence-BERT [15,16] that achieves state-of-the-art sentence embedding generation in several languages.

⁶ <https://www.qwant.com/?t=news>

Table 1. Proxy task regression performance for each model (average cosine similarity loss between the predicted embedding and the reference text embedding, lower is better).

Model	split by webpages			split by websites		
	train	valid.	test	train	valid.	test
distance	0.587	0.589	0.587	0.618	0.479	0.609
wGCN	0.303	0.342	0.339	0.357	0.355	0.417
GCN	0.365	0.407	0.406	0.426	0.482	0.569
GAT	0.367	0.441	0.439	0.423	0.563	0.623
DGCN	0.362	0.492	0.493	0.366	0.618	0.697

Table 2. WICE performance (average cosine similarity loss between the context node embedding and the reference text embedding, lower is better).

Model	split by webpages			split by websites		
	train	valid.	test	train	valid.	test
oracle	0.293	0.297	0.293	0.334	0.264	0.259
random	0.780	0.779	0.779	0.792	0.736	0.800
title	0.834	0.835	0.833	0.834	0.861	0.814
text after image	0.671	0.672	0.670	0.701	0.571	0.705
blind WICE	0.654	0.658	0.653	0.715	0.508	0.680
wGCN	0.381	0.386	0.381	0.415	0.404	0.441

Node types are also considered useful and are one-hot encoded into 22 groups based on their HTML tag’s semantics (lists, headers, paragraphs, etc.). There are two ways to split the dataset: splitting per document regardless of the original website and split by website using all the pages of one website. The second is more difficult because the data is not homogeneous: the test data may differ in structure and topic. In the first setting, the ratio of webpages in the training set, validation set and test set are 5:2:3, *i.e.* 59 775 webpages in 723 websites for the training set, 23 910 webpages in 622 websites for the validation set and 35 865 webpages in 677 for the test set. In the second setup, the proportions of websites in the training set, validation set and test set are also 5:2:3, *i.e.* 53 978 webpages from 402 websites in the training set, 28 872 webpages from 162 websites in the validation set and 36 700 webpages from 242 websites in the test set. The optimal cosine similarity regression losses for each model of the two settings are shown in Table 1. As can be seen, the explicit weight-GCN model performs better on both settings and generalizes significantly better on unknown websites. We use only the wGCN architecture for the WICE task.

The average cosine similarity loss between the proposed text, which is the text with the highest score, and the image’s reference text is shown in Table 2. We see that naive WICE heuristics mostly fail on such a diverse dataset: `title` performs even worse than `random`, while `text after image` (which can be view as window-based WICE) rarely picks the best text node. In theory, older works

managed WICE using more complex heuristics. However, defining a comprehensive ruleset does not scale up to more than 1000 websites and is unpractical in real applications. We do not compare with visual-based WICE, either, because the rendering step using requires at least 1 second per webpage⁷, *i.e.*, more than three days for our whole dataset, not even including the segmentation algorithm. The preprocessing of our approach, *i.e.* generating text embeddings and graphs, takes ≈ 0.429 second per webpage, totaling ≈ 21 hours.

For comparison, we add a random model that randomly choose a text node in the HTML page. Any model that has learnt anything should be better than this baseline.

The result shows that our model significantly outperforms WICE based on heuristics by at least 25%. The wGCN-extracted context is closer to the lower bound (`oracle`) than any other WICE approach we considered, thus validating our approach’s relevance. In practice, we found that a cosine similarity ≥ 0.6 (*i.e.* a cosine loss ≤ 0.4) is enough, considering that two sentences have the same topic. Our wGCN does not always reach this threshold in average but is significantly closer than other approaches that average around 0.3 similarity.

We found a strong correlation between regression losses and WICE losses during the exploration of the results (Pearson correlation coefficient: 0.96), suggesting that better models for proxy tasks are better models for the main task. We also found that texts with lower WICE losses are often semantically very close to the images and the reference texts or their topics. This observation may be summarized as a correlation between lower regression loss and the relevance of the image for the extracted text (the original WICE problem’s objective). This also shows that semantics may help in selecting nodes with similarly-named entities or dates, thus having a better chance to describe a given image.

We also observed that the model sometimes generalizes poorly to unknown websites, because of the heterogeneous webpages of each set.

However, this is less of a problem for closed set news crawler which parses a list of known websites regularly updated, with only the occasional introduction of a new domain.

4 Conclusion

In this work, we address the WICE problem by modeling and learning webpages using language models and GNNs, making large-scale automatic WICE easier, and not bounded by hard-coded rules. We train a model on a large unlabeled news webpage corpus by learning to mimick the alt-text when it exists. Our weight-GCN model assigns a weight to each text node and we then extract the most important node and define it as the image context. This approach may blindly extract context sentences using semantic similarity between sentences and structural information learned from the DOM tree. By working directly with the HTML, we avoid the need for rendering the webpage into an image and cut the preprocessing time by a factor 3, making large-scale WICE more tractable.

⁷ With *e.g.*, headless Chromium.

References

1. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: VIPS: a vision-based page segmentation algorithm. Tech. Rep. MSR-TR-2003-79, Microsoft (November 2003), <https://www.microsoft.com/en-us/research/publication/vips-a-vision-based-page-segmentation-algorithm/>
2. Coelho, T.A.S., Calado, P.P., Souza, L.V., Ribeiro-Neto, B., Muntz, R.: Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering* **16**(4), 408–417 (2004). <https://doi.org/10.1109/TKDE.2004.1269666>
3. De Cao, N., Aziz, W., Titov, I.: Question answering by reasoning across documents with graph convolutional networks. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 2306–2317. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1240>, <https://www.aclweb.org/anthology/N19-1240>
4. Fauzi, F., Hong, J.L., Belkhatir, M.: Webpage segmentation for extracting images and their surrounding contextual information. In: *Proceedings of the 17th ACM International Conference on Multimedia*. p. 649–652. MM '09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1631272.1631379>, <https://doi.org/10.1145/1631272.1631379>
5. Feng, H., Shi, R., Chua, T.S.: A Bootstrapping Framework for Annotating and Retrieving WWW Images. In: *Proceedings of the 12th Annual ACM International Conference on Multimedia*. p. 960–967. Multimedia '04, Association for Computing Machinery, New York, NY, USA (2004). <https://doi.org/10.1145/1027527.1027748>, <https://doi.org/10.1145/1027527.1027748>
6. Gong, Z., Uu, R.L.H., Cheang, C.: Web image indexing by using associated texts. *Knowl. Inf. Syst.* **10**, 243–264 (08 2006). <https://doi.org/10.1007/s10115-006-0011-0>
7. Hattori, G., Hoashi, K., Matsumoto, K., Sugaya, F.: Robust web page segmentation for mobile terminal using content-distances and page layout information. In: *Proceedings of the 16th International Conference on World Wide Web*. p. 361–370. WWW '07, Association for Computing Machinery, New York, NY, USA (2007). <https://doi.org/10.1145/1242572.1242622>, <https://doi.org/10.1145/1242572.1242622>
8. Joshi, P.M., Liu, S.: Web document text and images extraction using DOM analysis and natural language processing. In: *Proceedings of the 9th ACM Symposium on Document Engineering*. p. 218–221. DocEng '09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1600193.1600241>, <https://doi.org/10.1145/1600193.1600241>
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, Toulon, France (2017), <https://openreview.net/forum?id=SJU4ayYgl>
10. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9267–9276 (2019)
11. Li, G., Xiong, C., Thabet, A., Ghanem, B.: DeeperGCN: All you need to train deeper GCNs. *Computing Research Repository arXiv:2006.07739* (2020)
12. Li, J., Liu, T., Wang, W., Gao, W.: A broadcast model for web image annotation. In: Zhuang, Y., Yang, S.Q., Rui, Y., He, Q. (eds.) *Advances in Multimedia Information*

- Processing - PCM 2006. pp. 245–251. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
13. Lockard, C., Shiralkar, P., Dong, X.L.: OpenCeres: When open information extraction meets the semi-structured web. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3047–3056. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1309>, <https://www.aclweb.org/anthology/N19-1309>
 14. Qian, Y., Santus, E., Jin, Z., Guo, J., Barzilay, R.: GraphIE: A graph-based framework for information extraction. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 751–761. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1082>, <https://www.aclweb.org/anthology/N19-1082>
 15. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Asia World Expo, Hong Kong, China (11 2019), <https://arxiv.org/abs/1908.10084>
 16. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. CoRR **abs/2004.09813** (2020), <https://arxiv.org/abs/2004.09813>
 17. Shen, H.T., Ooi, B.C., Tan, K.L.: Giving meanings to WWW images. In: Proceedings of the Eighth ACM International Conference on Multimedia. p. 39–47. MULTIMEDIA '00, Association for Computing Machinery, New York, NY, USA (2000). <https://doi.org/10.1145/354384.376098>, <https://doi.org/10.1145/354384.376098>
 18. Tryfou, G., Tsapatsoulis, N.: Extraction of web image information: Semantic or visual cues? In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) Artificial Intelligence Applications and Innovations. pp. 368–373. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
 19. Tsapatsoulis, N.: Web image indexing using wice and a learning-free language model. In: Iliadis, L., Maglogiannis, I. (eds.) Artificial Intelligence Applications and Innovations. pp. 131–140. Springer International Publishing, Cham (2016)
 20. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. 6th International Conference on Learning Representations (2017)
 21. Wood, L., Le Hors, A., Apparao, V., Byrne, S., Champion, M., Isaacs, S., Jacobs, I., Nicol, G., Robie, J., Sutor, R., et al.: Document object model (dom) level 1 specification. W3C recommendation **1** (1998)
 22. Xie, S., Lu, M.: Interpreting and understanding graph convolutional neural network using gradient-based attribution methods. Computing Research Repository **arXiv:1903.03768** (2019), <http://arxiv.org/abs/1903.03768>