Information Retrieval Journal (2023) 26:11
https://doi.org/10.1007/s10791-023-09424-3

Heterogeneous graph attention networks for passage retrieval

Lucas Albarede[1,2] · Philippe Mulhem[1] · Lorraine Goeuriot[1] · Sylvain Marié[2] · Claude Le Pape-Gardeux[2] · Trinidad Chardin-Segui[2]

Received: 19 August 2022 / Accepted: 6 September 2023 / Published online: 16 November 2023

Abstract
This paper presents an exploration of the usage of Heterogeneous Graph Attention Networks, or HGATs, for the task of Passage Retrieval. More precisely, we study how these models perform to alleviate the problem of passage contextualization, that is incorporating information about the context of a passage (its containing document, neighbouring passages, etc.) in its relevance estimation. We first propose several configurations to compute contextualized passage representations, including a document graph representation composed of contextualizing signals and judiciously modified HGAT architectures. We then present how we integrate these configurations in a neural passage ranking model. We evaluate our approach on a Passage Retrieval task on patent documents: CLEF-IP2013, as these documents possess several different contextualizing signals fully exploited in our models. Our results show that some HGAT architecture modifications allow for a better context representation leading to improved performances and stability.

Keywords Graph attention network · Learning · Negative sampling · Evaluation · Passage retrieval · Document representation

✉ Lucas Albarede
lucas.albarede@protonmail.com

Philippe Mulhem
Philippe.Mulhem@imag.fr

Lorraine Goeuriot
lorraine.goeuriot@univ-grenoble-alpes.fr

Sylvain Marié
sylvain.marie@se.com

Claude Le Pape-Gardeux
claude.lepape@se.com

Trinidad Chardin-Segui
trinidad.chardin-segui@se.com

[1] Univ. Grenoble Alpes, Centre National de la Recherche Scientifique, 38000 Grenoble, France

[2] Schneider Electric, 160 Av. des Martyrs, 38000 Grenoble, France

1 Introduction

Passage Retrieval is an Information Retrieval (IR) topic concerned with retrieving small textual elements. A key problem rising from this task is properly estimating the relevance of an element which is most of the time an excerpt of a longer document. To alleviate this issue, several Passage Retrieval approaches resort to contextualization (Albarede et al. (2021); Bendersky and Kurland (2008); Callan (1994); Fernández et al. (2011); Murdock and Croft (2005); Sheetrit et al. (2019); Albarede et al. (2022)); that is, the consideration of a passage's context in its relevance estimation. Such approaches exploit the relations between a passage and different signals coming from its environment to better estimate its similarity to a query.

Several types of signals have been exploited in the literature to contextualize a passage such as the content of its document, the content of other passages in the same document or the document's structure. However, most works on contextualization focus on exploiting one or two signals while discarding others even when available

A recent work showed that several signals can be condensed into a document graph representation where they act as the relations (edges) between passages (nodes) (Albarede et al. (2022)). Moreover this work exploits Graph Attention Networks, methods that represent the content of nodes in a graph with respect to the relations between them in order to compute, for each passage in the document graph representation, a contextualized representation.

We extend this work by considering abstract document representations that can be formed by any signals in conjunction with modified GAT architectures to derive several passage ranking models.

More specifically and with the passage retrieval task in mind, we theorize on some desirable behaviours of these models and modify the architecture of Heterogeneous Graph Attention Networks (HGATs) accordingly. Then we define several passage ranking models, each exploiting a different modified HGAT architecture.

We perform experiments on the CLEF-IP2013 passage retrieval task which is tasked with retrieved excerpts of patent documents. Such dataset is adequate to our approaches since the documents it is composed of contain several different contextualization signals. Our findings show that a classical unmodified HGAT is not suited for the passage retrieval task and that judicious modifications lead to significant performance improvements.

This paper is organized as follows. In Sect. 2 we present a state-of-the-art study on the passage contextualization problem and the use of graph neural network in IR, before presenting our graph-related propositions in Sect. 3 and how we integrate them in passage ranking models in Sect. 4. We define our experimental setup in Sect. 5 and discuss the results of our experiments in Sect. 6, before concluding.

2 Related works

2.1 Signals exploited for contextualization

Passage contextualization is a long lasting problem for the Passage Retrieval task. We provide an overview of the different signals studied in the literature as well as how they are exploited for passage contextualization. We note that as the definition of passage varies from one work to another, we consider here a passage as any textual excerpt of document.

A common way to contextualize a passage is by considering its containing document. In this case, the document is considered as a unit and approaches exploit it by either computing document-query similarities (Sheetrit et al. (2019); Bendersky and Kurland (2008)) or by using its language model to smooth the passage's language model (Murdock and Croft (2005); Callan (1994); Fernández et al. (2011)). More recently some works have used this signal to improve neural retrieval, either considering the document's title (Lu, Ábrego, Ma, Ni and Yang (2020)) or by using a powerful document retrieval step in a re-ranking setup (Nogueira and Cho (2019)).

Other studied ways to contextualize a passage is the consideration of other passages from the same document. Passages of a document are represented as a list according to their order of appearance and a passage is contextualized using its neighbours, the intuition being that closer passages are better contextualizing signals. For instance, some works only consider the previous and following passages (Sheetrit et al. (2019); Wu et al. (2022); Chen et al. (2022)) while others contextualize a passage with respect to its distance to other passages (Carmel et al. (2013); Fernández et al. (2011); Krikon et al. (2011); Albarede et al. (2021)). More recently, neural functions such as convolution layers (Hofstätter, Mitra, Zamani, Craswell and Hanbury (2021)) or recurrent neural layers (Arnold et al. (2020)) have been used to give a passage a representation with respect to its neighbourhood.

Structural signals have been mostly exploited in the Structured Retrieval task: exploiting the graph structure of a document in order to retrieve smaller structural elements. Works have shown that contextualization is of high importance in such task (Kekäläinen, Arvola, and Junkkari (2018)) and an effective way of performing contextualization is the notion of structural neighbourhood contextualization (Norozi and Arvola (2013); Norozi , Arvola and de Vries (2012); Arvola et al. (2005, 2008); Albarede et al. (2021)): propagating a part of document's relevance in an uniform manner across its neighbourhood in order to contextualize other parts of document. Such an idea has been developed to take into account the distance between two elements in the relevance propagation (Albarede et al. (2021)). Other works exploit the tree structure of documents and argue that a structural element's score should depend on the scores of its children (Callan (1994); Kaszkiel et al. (1999); Ogilvie and Callan (2005); Mass and Mandelbrod (2005)). Furthermore, they use these new founded scores to smooth down the relevance of their children. Concerning neural retrieval, a recent work has studied the propagation of a document's title name alongside its structure to improve contextualization (Liu et al. (2021a)).

References between documents (or parts of documents) have been studied for the passage contextualization task, mostly because such signals are very corpus dependent. However, some works have studied the use of random walks on citation graphs in order to estimate the importance of document parts in the corpus (Norozi , Arvola and de Vries (2012); Norozi and Arvola (2013); Norozi, de Vries and Arvola (2012)).

All of the aforementioned contextualizing signals can be embedded with document parts in a graph, as highlighted in a recent work (Albarede et al. (2021)). We extend this work and propose a passage contextualization method that is signal independent, that is, that can be applied with any kind and number of contextualizing signals as long as they can be represented in a graph.

2.2 Graph neural networks in information retrieval

Graph Neural Networks (GNNs) are methods computing the representation of a node in a graph with respect to its neighbours. They have been successfully applied in several

Information Retrieval tasks to improve the representation of documents. For instance some works use these methods on word graphs (Xie et al. (2021); Cui et al. (2022); Qi et al. (2020)), where nodes represent words in a document an edges represent their co-occurrences. This leverages their interactions and allows for a better representation of the document's content. Others works have applied GNNs on other kind of graphs such as a query - document graph during training (Li et al. (2020)) or a graph linking documents to the entities they contain (Zhao et al. (2019)).

Graph attention networks, a specific type of graph neural networks introducing a notion of attention between nodes, have also been studied in an Information Retrieval context. Similarly to previous works, it has been used to compute document representations from graphs composed of words (Zhang et al. (2018)). These methods have also been studied in other contexts. For example, GATs have been used in a web-scale retrieval setup to leverage user co-click information between documents (Zhang et al. (2021)), or in a cross-modal retrieval task to bridge the gap between video and text representation (Hao et al. (2021)).

Closer to our approach, some works have used graph attention networks on graphs composed of passages, either exploiting the structure of their containing document to refine its representation (Xu et al. (2021)), or by linking passages to queries during training to produce query-interactive passage embeddings (Liu et al. (2022)).

Similar to these works, we use GNNs (specifically HGATs) to compute node representations relative to their neighbours in a passage retrieval setup. Following a recent work (Albarede et al. (2021)), we focus on two key points: **(1)** First, we represent not only full documents but any parts contained in a document. **(2)** Second, we exploit graphs composed of document parts and their relations. However, we study how judicious modification of the HGAT architecture affects its representation capability.

3 Graph-related propositions

Our main objective is to perform passage contextualization using HGAT models. More precisely we exploit these models, alongside a graph representing the document elements and their relations, to compute contextualized passage representations. In this section we first present our definition of a document graph representation, display an overview of classical HGAT models and then see how we modify them to better suit our needs.

3.1 Document graph representation

This work relies on graphs that represent the passages and their relations with the different elements in the corpus. Since our approach is not dependent on the different signals exploited to build the graphs, we define a document graph representation as a directed graph $G = (V, E, A, R, \tau, \phi)$ where V is the set of nodes, E the set of edges, A the set of node types, R the set of edge types, τ the node type mapping functions $\tau : V \rightarrow A$ and ϕ is the edge type mapping functions $\phi : E \rightarrow R$.

An edge $e \in E$ is defined between two nodes $v, w \in V$ and we define the set of edges from v to w as e_{vw} such that $\forall e' \in e_{vw}, \phi(e')$ is unique.

Typically, A contains document part types such as sections, chapters, passages, etc. On the other hand, R contains any signal types between document parts such as references, structural relations, etc.

3.2 Classical HGAT

Graph attention networks are multi-layer graph neural networks which compute an embedding for each node in a graph by taking into account information from its neighbours (Veličković et al. (2017)). Each layer aggregates, for each node, its embedding with the embedding of its neighbours using attention functions (Bahdanau et al. (2014)). Stacking n layers allows a node to gather information about nodes that are at a distance of n hops in the graph. One element worth mentioning is that GATs implicitly add *self-edges* connecting each node to itself to build the embedding of a node.

HGATs are GATs modified to take into account the plurality of nodes and edges types by modeling their differences with different attention functions and separate weights (Wang et al. (2019)).

In this work, we use *Attention is all you need* (Vaswani et al. (2017)) definition of attention and define one attention function $MultiHead_r$ per type of edge r, so the model treats the interaction between nodes differently according to the type of their relation (Wang et al. (2019)). For each r, $MultiHead_r$ is a function of three variables representing the queries, keys and values modeling the attention mechanism. The model defines a learnable weight vector W_r for each type of edge in the graph, representing the global importance of the relation. For a node i, its neighbour nodes N_i and e_{ij} the set of edges between nodes i and j (with j in N_i), the representation of i is computed as:

$$h'_i = \sum_{j \in N_i} \sum_{e' \in e_{ij}} softmax(W_{\phi(e')}) * MultiHead_{\phi(e')}(h_i, h_j, h_j) \tag{1}$$

3.3 Architecture modifications

Classical HGAT implementation relies on several elements that may be questioned in the case of passages retrieval according to the state of the art findings:

F1. As described earlier in Sect. 3.2, a node propagates indirectly its initial content to itself. This feature may not be desirable as we would like to precisely control what propagation is activated during the passage retrieval.

F2. The weights learned for each relation in the network are all independent. However, previous works have shown that explicitly considering content and context information separately is beneficial for passage contextualization.

F3. By default, the initial content of a target node is included in the computation of its in-context representation. It is then difficult to assess to which extend this content is accurately considered.

We define different constraining modifications that alleviate potential problems coming from the aforementionned elements. Unlike the classical HGAT models, some of these modifications focus on the computation of the representation of a specific node in the graph. In order to better describe them, we introduce the notation *tgt* to designate the target node of a HGAT representation computation.

No out modification:

In order to solve the problem highlighted in **F1.**, we define the *No out* modification. This modification prevents the model from propagating the target node's representation to its neighbours by simply not considering any of its non-self out-going links. We think that this might allow the model to better learn to merge the target node's content and context information. The HGAT function is modified as such:

- To compute the intermediate representation of the target node, we use equation (1) above.
- To compute the intermediate representation of every other node, we use equation (2):

$$h'_i = \sum_{j \in N_i, j \neq tgt} \sum_{e' \in e_{ij}} softmax(W_{\phi(e')}) * MultiHead_{\phi(e')}(h_i, h_j, h_j) \tag{2}$$

Lambda modification:

In order to solve the problem highlighted in **F2.**, we define the *Lambda* modification. As in the state of the art shown that weighting on one side the content and on the other side all the relationships, we study the usage of adding a global weight applied on all the non-self relations. We introduce a different parameter λ in each layer that forces the model to take into account content and context as two distinct elements. We do not constrain this modification to the computation of the target node only since we think that it might improve the overall HGAT modeling capacity.

The HGAT function is modified as such:

$$\begin{aligned} h'_i =& (1 - \lambda) * MultiHead_{\phi(e_{ii})}(h_i, h_i, h_i) \\ &+ \lambda * \Big(\sum_{j \in N_i, j \neq i} \sum_{e' \in e_{ij}} softmax(W_{\phi(e')}) * MultiHead_{\phi(e')}(h_i, h_j, h_j) \Big) \end{aligned} \tag{3}$$

No self modification:

In order to solve the problem highlighted in **F3.**, we define the *No self* modification. We explore a modified version of the GATs that does not at all consider the target node initial content when computing its final context-based content. To do that, we simply omit the self-link of the target node when computing its representation.

The HGAT function is modified as such:

- To compute the intermediate representation of the target node, we use equation (2).
- To compute the intermediate representation of every other node, we use equation (1).

3.4 Resulting HGAT variant

We study the impact of each of the modifications independently as well as when combined with each other. We define eight different HGAT architectures corresponding to the different possible combinations. General information about these architectures is reported in Table 1, and a detailed function description is given in Appendix 1. The first three columns represent the three modifications and the last column the name of the HGAT variants that are going to use in the following of this work. Each line is a different variant and a check mark in a cell indicates that the variant is created with the according modification. For example, the name LB_NS means that the variant is created from a classical HGAT model on which is applied the modifications ***Lambda*** (LB) and ***No self*** (NS).

Table 1 Definition of the HGAT variants. A check mark indicates that the variant considers the according architecture modification

No Out	Lambda	No Self	HGAT variant name
			BASE
✓			NO
	✓		LB
		✓	NS
✓	✓		NO_LB
✓		✓	NO_NS
	✓	✓	LB_NS
✓	✓	✓	NO_LB_NS

4 Passage ranking models

Our objective is to exploit HGATs to perform passage ranking, that is computing the similarity between a passage and a query. More precisely, our idea is to compare the contextualized passage representations computed by these models to a query representation. We discuss in this section how we integrate the HGAT variants defined in Sect. 3.4 into passage ranking models. We first present two passage ranking frameworks designed to compute passage representations with HGATs, present some of their shared components and finally see how we combine our variants and the ranking frameworks to yield the final passage ranking models.

4.1 Ranking frameworks

We derive a "standard" passage ranking framework as well as a "late fusion" framework, specifically designed to make up for a major flaw in two of our HGAT variants.

Standard framework

The idea behind the standard Passage Ranking framework is that it estimates a passage relevance to a query by first computing an *in-context* passage representation $E_{in-context_p}$ by feeding the passage and its neighbours encoded representations to a HGAT model, and then estimating its similarity (using equation (6)) to the encoded representation of the query E_q. The framework is summarized in Fig. 1 and equation (4) below:

$$relevance(q, p) = sim(E_q, E_{in-context_p}) \tag{4}$$

Late fusion framework

The *in-context* passage representation used in the standard framework assumes that it contains both passage content and passage context information. However some HGAT variants presented in Sect. 3.4, namely NO_NS and NO_LB_NS, remove both the target node's outgoing connections as well as its self connection. This causes the target node's representation computed by the HGAT to omit content information and only contain context information. Previous works on passage contextualization have shown that passage content information is required in order to perform effective passage retrieval (Albarede et al. (2021); Bendersky and Kurland (2008); Callan (1994); Fernández et al. (2011);

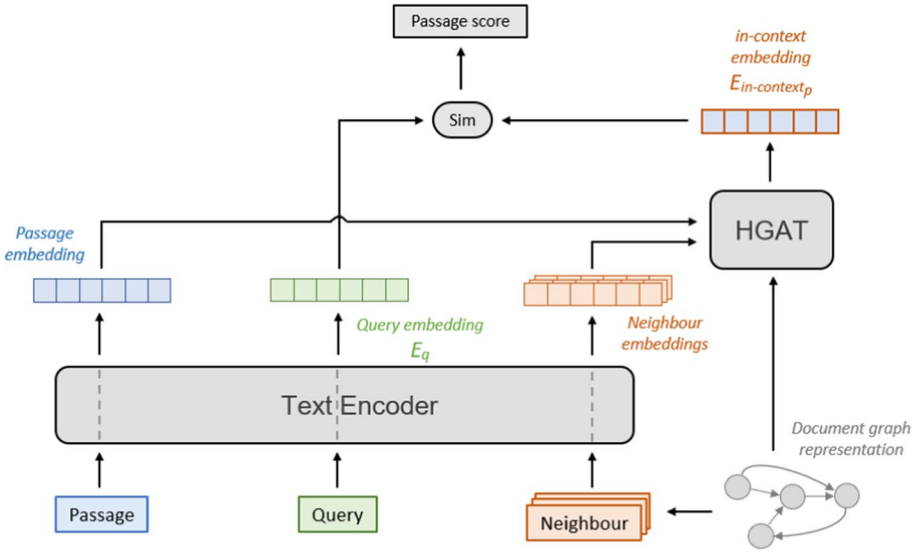

Fig. 1 Overview of the standard ranking framework

Murdock and Croft (2005); Sheetrit et al. (2019); Albarede et al. (2022)). In order to inject passage content information into the passage retrieval pipeline, we adapt the ranking framework with a late fusion mechanism, specifically designed to use HGAT variants NO_NS and NO_LB_NS. Similar to (Albarede et al. (2021, 2022)), the late fusion framework computes two passage-query similarities: **(i)** a first one with a *context-only* passage representation $E_{context-only_p}$ computed by the HGAT variant, **(ii)** and a second one with a *content-only* passage representation E_p computed by the encoder. To obtain the final passage relevance, the two query similarities are then linearly combined using a parameter α. The framework is summarized in Fig. 2 and equation (5) below:

$$relevance(q,p) = (1-\alpha) * sim(E_q, E_p) + \alpha * sim(E_q, E_{context-only_p}) \tag{5}$$

4.2 Framework components

Encoder

We use the multiple representation text-encoder taken from the state-of-the-art ColBERT model (Khattab and Zaharia (2020)) to embed text into a dense semantic space. Instead of encoding all possible text information into a single embedding (usually corresponding to the special token [CLS]), each term's embedding encodes its contextualized semantic information. Despite being less efficient in terms of response time and memory usage, these representations have shown to outperform the classical single representation approach across several IR metrics (Macdonald et al. (2021); Santhanam et al. (2022)).

Similarity measures

The similarity function we use, described in (Khattab and Zaharia (2020)), exploits the modelling capability of multiple representation embeddings. More precisely, every query

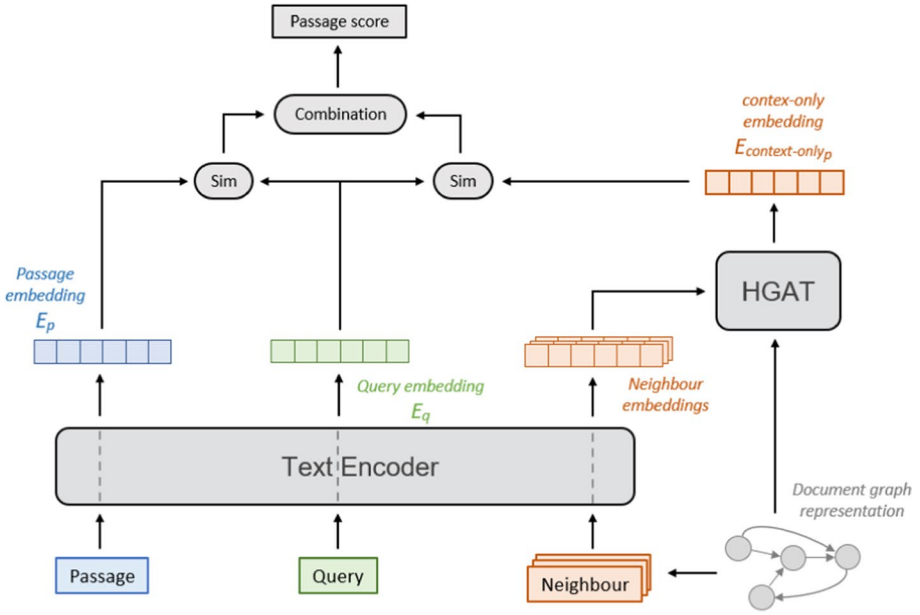

Fig. 2 Overview of the late fusion ranking framework

embedding interacts with all passage embeddings by computing the maximum cosine similarity, and these maxima are summed across query terms. The similarity between E_q and E_p, the multiple representation embeddings from a query q and a passage p is:

$$sim(E_q, E_p) = \sum_{i \in [1, \|E_q\|]} \max_{j \in [1, \|E_p\|]} E_{q_i} \cdot E_{p_j}^T \tag{6}$$

This similarity function has the advantage of being computationally light and thus does not hinder the retrieval execution time. We do not consider more computationally expensive ways to combine embeddings and leave this for future works.

4.3 Final ranking models

In order to create a final ranking model, we integrate a HGAT variant in a passage ranking framework. This is done by simply replacing the HGAT component in the framework with the desired HGAT variant. As stated earlier, we combine HGAT variants NO_NS and NO_LB_NS with our late fusion framework and every other variants with the standard framework. Table 2 presents the different ranking models. Each ranking model is named after the framework and HGAT variant it is composed of. For example, the *std_NO_LB* model is composed of the *standard* framework and the *NO_LB* HGAT variant.

Table 2 Final ranking models. Each model is composed of a ranking framework associated with a HGAT variant

Framework	HGAT variant	Final ranking model
Standard	BASE	*std_BASE*
	NO	*std_NO*
	LB	*std_LB*
	NS	*std_NS*
	NO_LB	*std_NO_LB*
	LB_NS	*std_LB_NS*
Late fusion	NO_NS	*late_NO_NS*
	NO_LB_NS	*late_NO_LB_NS*

5 Experimental setup

In this section, we first present the experimental resources such as the dataset description, the query and the passages descriptions. Then, we focus on the experimental process discussing technical details, how we train our models and how we implement them in a Passage Retrieval setup.

5.1 Experimental resources

Even though our models can be used on documents containing any kind of contextualizing signals (see Sect. 3.1), our objective in this work is to study how they behave when exploiting several different signals at the same time. We evaluate our approaches on CLEF-IP, a corpus composed of 2.6 million patents as these documents are structured, can be decomposed into passages, and possess intra- and inter-document references.

5.1.1 Passage retrieval task information

Specifically, our models are evaluated on the CLEF-IP2013 passage retrieval task (Piroi et al. (2013)). The task contains French, German and English queries separated in train and test sets. We conduct our experiments on English queries only, which amounts to 56 training queries and 50 test queries.

CLEF-IP-2013 uses five evaluation measures. To be compliant with the original tasks, we report here the same measures. A relevant document is a document which contains at least one relevant passage. Three measures are computed at the document-level: **(1)** PRES@100 which measures the effectiveness of ranking documents relative to the best and worst ranking cases, the best case being that all relevant documents are retrieved at the top of the list, and the worst being that all relevant documents are retrieved just after the maximum number of documents to be checked by the user[1] (in this case, 100), **(2)** RECALL@100 and **(3)** MAP@100. Two measures are computed on the passage level: **(4)** MAP(D) which computes the AP inside each relevant document (with respect to its passages), averages this score for a query over its relevant documents, and averages it across

[1] http://homepages.inf.ed.ac.uk/wmagdy/PRES.html.

all queries to get the MAP. **(5)** PREC(D) which computes the precision inside each relevant document and averages the scores in the same manner as for MAP(D).

5.1.2 Queries construction

The objective of the CLEF-IP2013 retrieval task is *prior art search*: finding passages of patents that are similar to one or several query claims coming from a patent document. It is a popular practice to use such full patent document as a source and to transform it into a short, refined query (Mahdabi et al. (2011); Xue and Croft (2009); Mahdabi et al. (2013); Andersson et al. (2016)). For all our experiments, we use the state-of-the-art method (Mahdabi et al. (2011)): let q_d be a query patent document with a set of claims, we build a first form of the query with the top-10 words with highest *tf-idf* in q_d's abstract and a second form composed of q_d's natural language claims. The usage of the two query forms is specified below in Sect. 5.2.2.

5.1.3 Patent-specific document graph representation

Patents are hierarchically structured documents containing intra- as well as inter-document references. We consider several relations that have been leveraged in the past in order to build the patent-specific document graph representation. The *intra*-document relations considered are: **(1)** the *order* of passages (Beigbeder (2010); Fernández et al. (2011); Krikon et al. (2011); Sheetrit et al. (2019)), **(2)** the *hierarchical structure* of the document (Albarede et al. (2021); Norozi and Arvola (2013); Norozi , Arvola and de Vries (2012)) and **(3)** *internal* references. One *inter*-document relation is considered: the **(4)** *external* reference of one document by another one. As in (Norozi , Arvola and de Vries (2012); Norozi, de Vries and Arvola (2012)), we also include the *inverse* relation, denoted by the subscript $_i$, of each of these relations.

In order to consider these relations, we define two types of nodes: *passage* nodes that represent textual units and *section* nodes that represent titled structural units – and eight types of edge (one for each relation and its symmetrical): *order* characterizing the relation order between *passage* nodes ($order_i$ its symmetrical), *structural* characterizing the composition between a *passage* node and a *section* node or between two *section* nodes ($structural_i$ its symmetrical), *internal* characterizing the *intra*-document references between nodes ($internal_i$ its symmetrical) and *external* characterizing the *inter*-document references between nodes ($external_i$ its symmetrical). Symmetrical relations are added in order to let a targeted node access information about the source node in order to increase the contextualizing capacity of the representation.

According to our previous notations of Sect. 3.1, we have $A = \{passage, section\}$ and $R = \{order, order_i, structural, structural_i, internal, internal_i, external, external_i\}$.

Figure 3 shows our process of transforming a document into its graph representation. Circles represent *sections* and squares represent *passages*. Inverted edges in the final graph representation are drawn as highlighted.

Relation extraction

In order to build the *structural* relations, we extract the structure from the patent documents. They are XML documents most of the time segmented in four main sections: bibliography, abstract, description and claims. We use these four sections as starting points in the XML structure to look for other sections using hand-crafted features, either based on XML tags, case or number of characters.

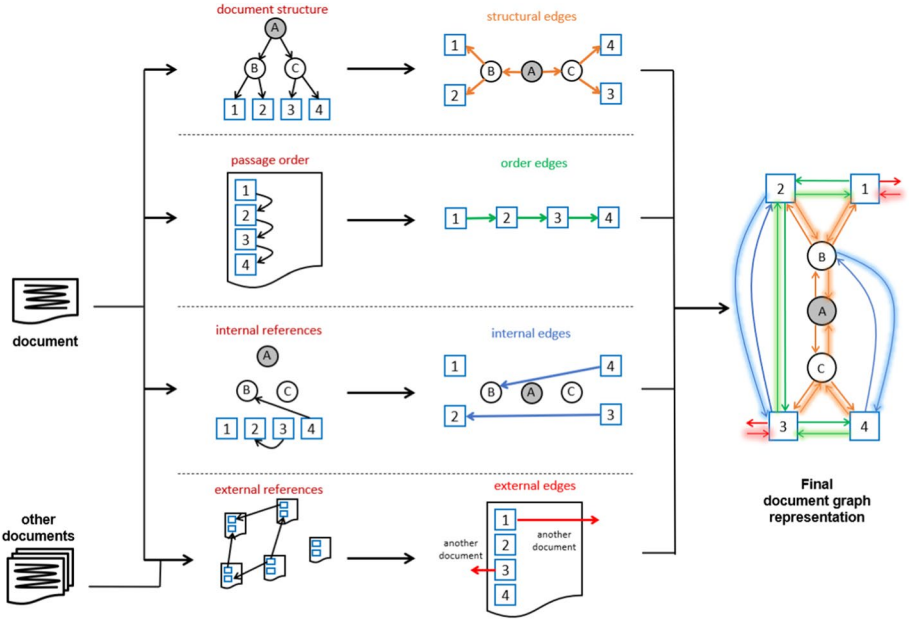

Fig. 3 Transformation of a document (left hand-side) into its graph final representation

In order to build the *internal* relation, we find references to patent claims in the text. *external* reference relations are built using the patent citation tags.

5.2 Experimental process

5.2.1 Technical details

We use the pyterrier IR system (Macdonald and Tonellotto (2020)) alongside the Pytorch framework (Paszke et al. (2019)) during our experiments. Text embeddings are encoded from the state-of-the-art ColBERT model trained on MSMARCO (Nguyen et al. (2016)), with a maximum passage length of 180 tokens and a maximum query length of 120 tokens. Our HGAT variants are composed of 3 layers, each layer having several attention functions $MultiHead_r$ with 8 heads using dropout=0.7. Each learnable weight in a HGAT layer is a vector of size 180.

We use standard Pytorch initialisation for every $MultiHead_r$ function and randomly initialise λ and α between [0, 1].

5.2.2 Retrieval process

As is often done in neural passage retrieval, we place ourselves in a re-ranking setup: we first retrieve a list of passages using a classical retrieval model before re-ranking them using our final ranking models. More precisely, for a given query, we use the BM25 model (Robertson et al. (1995)) (with Pyterrier (Macdonald and Tonellotto

(2020)) default parameters) to retrieve the top-1000 documents using the first query form, before ranking every passage contained in these documents using the second query form.

During the document retrieval step, we use a filtering mechanism to eliminate documents which do not share any IPC code (International Patent Classification: codes grouping patents according to different criteria) with the query patent document. Even if this filtering might discard relevant documents, it brings good performances in practice (Gobeill and Ruch (2012)).

Since our passage ranking approaches rely on creating query-independent passage representations, before calculating query similarities using a computationally light mechanism (equation (6)), it is possible to compute these representations offline and thus only using a look-up mechanism during re-ranking, vastly increasing efficiency.

5.2.3 Learning process

The Adam optimizer (Kingma and Ba (2014)) is used to jointly learn the HGATs parameters and to fine-tune the encoder. For the encoder, we use the advised (Khattab and Zaharia (2020)) learning rate of $3 * 10^{-6}$ and freeze the first six layers. For the graph-based model, the weight vectors W_r, the parameters λ and α, we use a learning rate of $1 * 10^{-3}$. Our learning process is as follows: given a triple $\langle q, p^+, p^- \rangle$ with query second form q, positive passage p^+ and negative passage p^-, the model is optimised via pairwise softmax cross-entropy loss over the computed scores of p^+ and p^-.

Negative Sampling

Negative sampling in triplet loss learning has shown to be an important task-specific factor in neural passage ranking effectiveness (Lu et al. (2020); Liu et al. (2021b); Lu et al. (2021); Cohen et al. (2019); Hong et al. (2022)). In our case, our ranking models consider both the content and the context of a passage to estimate its relevance. We hypothesize that, if we want them to properly learn how to merge these information, we need to confront them during learning with positive and negative passages having similar context.

To confirm that, we derive five different negative sampling strategies and study what are the most effective negative passages for a pair $\langle q, p^+ \rangle$:

- The *Random* strategy randomly samples a non-relevant passage from the corpus.
- The $Same_{doc}$ strategy randomly samples a non-relevant passage from p^+'s document.
- The $Relevant_{docs}$ strategy randomly samples a non-relevant passage from the set of q's relevant documents.
- The $Shared_{codes}$ strategy randomly samples a non-relevant passage from the set of documents that share an IPC classification code with q's patent document q_d.
- The *Mixed* strategy is a mix between the $Relevant_{docs}$ strategy and the $Shared_{codes}$ strategy. We randomly choose the $Relevant_{docs}$ with a probability of 0.4 and the $Shared_{codes}$ strategy with a probability of 0.6.

We repeat this process 1000 times, yielding for each sampling strategy $3.5M$ triplets.

Results presented in Sect. 6.1 will determine which sampling strategy to use for the main experiments.

Table 3 Results of our negative sampling experiments over CLEF-IP2013, given in percentage. Results are presented under the form mean ± st.dev. *i*, *j*, *k*, *l*, *m* design statistical significance over *Random*, $Same_{doc}$, $Relevant_{docs}$, $Shared_{codes}$ and *Mixed* respectively using a Mann–Whitney U test (p-value =0.05). Boldface indicates best mean result per column

Neg. Strat.	PRES@100	Recall@100	MAP@100	MAP(D)	PREC(D)
Random	$37.1^{j} \pm 2.1$	$45.5^{j} \pm 2.9$	$10.1^{j} \pm 2.7$	6.1 ± 2.2	7.8 ± 2.5
$Same_{doc}$	27.1 ± 0.3	37.7 ± 0.2	8.4 ± 0.8	5.1 ± 0.4	8.3 ± 0.3
$Relevant_{docs}$	$44.1^{ij} \pm 1.7$	$53.7^{ij} \pm 2.0$	$14.1^{ij} \pm 2.4$	$8.1^{ij} \pm 2.1$	$11.1^{ij} \pm 1.6$
$Shared_{codes}$	$43.6^{ij} \pm 1.8$	$52.6^{ij} \pm 2.5$	$13.8^{ij} \pm 2.8$	$7.8^{ij} \pm 2.2$	$10.5^{ij} \pm 2.2$
Mixed	$\mathbf{46.1}^{ijkl} \pm 1.8$	$\mathbf{56.4}^{ijl} \pm 1.9$	$\mathbf{15.7}^{ijl} \pm 2.2$	$\mathbf{9.2}^{ij} \pm 2.5$	$\mathbf{12.9}^{ijkl} \pm 2.2$

6 Experiments

We first present our preliminary experiments aimed at fixing the negative sampling strategy and then dive into our main experiments aimed at studying the effectiveness of HGAT variants for passage retrieval.

6.1 Negative sampling experiments

The effectiveness of the different negative sampling strategy is presented, using one ranking model: std_BASE. This is done to prevent potential biases coming from our architecture modifications during the preliminary experiments. For each negative sampling strategy, we learn and test the retrieval model 10 times to lessen the effect of the parameters' random initialisation.

We report these results in Table 3, under the form $mean \pm st.dev$ in percentage. A statistical significance analysis, with a Mann–Whitney U test with p-value of 0.05, is achieved between the 10 runs of every strategies. Such significance test is adequate since we do not have any prior on the distribution of this data, and the sample size is quite small.

Looking at the results, we see that the *Random* strategy performs significantly worse than three other strategies. This is an expected behaviour showing the importance of performing more sensible negative sampling.

Comparing the three strategies $Same_{doc}$, $Relevant_{docs}$ and $Shared_{codes}$, we see for instance that the latter two show similar results improving upon *Random* by respectively $7/37.1 = 18.8\%$ and $6.5/37.1 = 17.7\%$ for PRES@100. This support our hypothesis that sampling negative passages according to their context improves performances. However, we see that the $Same_{doc}$ strategy gives worse results than all other approaches. A potential explanation of this behaviour is that the set of passages sampled by this strategy is too small, and that the same passages are sampled several times, decreasing variability in the training triplets and thus increasing the chance of model over-fitting.

Finally, the last line of Table 3 shows that the *Mixed* strategy significantly outperforms $Relevant_{docs}$ and $Shared_{docs}$ on two of the four evaluation measures: PRES@100 and PREC(D). According to these results, we use then the *Mixed* negative sampling strategy for our main experiments.

Table 4 Results of our main experiments over CLEF-IP2013, given in percentage. Results are presented under the form mean ± stdev. *i*, *j*, *k*, *l*, *m*, *n*, *o*, *p* design statistical significance over *std_BASE*, *std_NO*, *std_LB*, *std_NS*, *std_NO_LB*, *std_LB_NS*, *late_NO_NS* and *late_NO_LB_NS* respectively using a Mann–Whitney U test (p-value =0.05). Boldface indicates best mean result per column

Model	PRES@100	Recall@100	MAP@100	MAP(D)	PREC(D)
std_BASE	$46.4^{ln} \pm 1.5$	$56.1^{j} \pm 2.3$	$16.1^{n} \pm 2.5$	$9.7^{l} \pm 1.5$	$13.7^{ln} \pm 1.9$
std_NO	$49.2^{ikln} \pm 1.0$	$58.7^{kln} \pm 2.0$	$21.3^{ikln} \pm 1.8$	$19.1^{ikln} \pm 1.7$	$23.1^{ikln} \pm 1.4$
std_LB	$46.1^{ln} \pm 1.4$	$54.7^{l} \pm 2.0$	14.8 ± 1.9	$11.8^{l} \pm 0.8$	$14.7^{ln} \pm 1.2$
std_NS	32.5 ± 3.5	41.7 ± 3.0	12.0 ± 3.0	8.4 ± 2.3	10.1 ± 2.6
std_NO_LB	$52.2^{ijkln} \pm 0.8$	$\mathbf{61.5}^{ijkln} \pm 1.8$	$23.9^{ijkln} \pm 1.5$	$21.9^{ijkln} \pm 0.9$	$26.3^{ijkln} \pm 0.5$
std_LB_NS	35.1 ± 3.4	44.1 ± 3.0	14.8 ± 3.0	$11.1^{l} \pm 2.4$	$12.3^{l} \pm 3.1$
late_NO_NS	$\mathbf{53.8}^{ijkln} \pm 2.9$	$61.3^{ikln} \pm 5.1$	$\mathbf{25.3}^{ijkln} \pm 4.9$	$\mathbf{23.1}^{ijkln} \pm 4.8$	$\mathbf{26.6}^{ikln} \pm 6.3$
late_NO_LB_NS	$53.4^{ijklmn} \pm 1.0$	$61.2^{ijkln} \pm 2.5$	$25.2^{ijkln} \pm 2.1$	$22.7^{ijkln} \pm 3.2$	$26.2^{ikln} \pm 4.5$

6.2 Main experiments

For our main experiments, as in the preliminary experiments, we learn and test every model ten times so as to study both their effectiveness and robustness. We report the main results in Table 4 under the form *mean* ± *st.dev*. We perform a Mann–Whitney U test with p-value of 0.05 as we previously did.

We study models effectiveness and robustness separately, before giving a parameter analysis and performance comparison with the state-of-the-art.

6.2.1 Models effectiveness

Focusing our effectiveness analysis on models using the standard framework first, we see in the first two lines of Table 4 that *std_NO* significantly increases performances over the base model, especially for MAP(D) and PREC(D). However, we see that *std_LB* yields similar results as *std_BASE*, and that *std_NS* significantly decreases the results in four evaluation measures. Finally, the *std_NO_LB* model significantly improves results over all other models that are built with the standard framework.

Looking at the late fusion models, we see that they yield very similar results, with *late_NO_NS* slightly ahead though.

Analyzing the results globally, *late_NO_NS* and *late_NO_LB_NS* give better results than the models built with the standard framework. More precisely, *late_NO_LB_NS* significantly outperforms most of them for four evaluation measures and all of them for the PRES@100 measure.

Takeaways focusing on our HGAT architecture modifications are that the NS modification only performs while implemented with the late fusion framework, the NO modification improves performances over the base model and by a larger margin when combined with the LB modification.

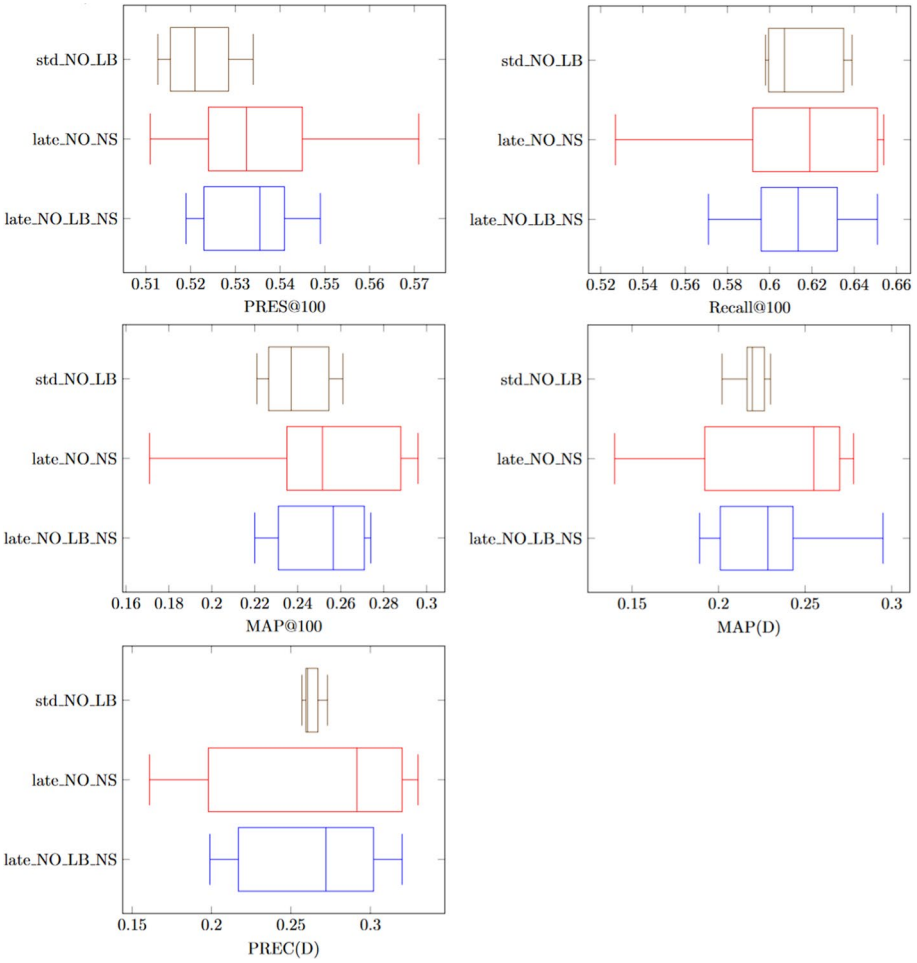

Fig. 4 Distribution of performance of three models on each evaluation measure. Each boxplot is composed of the minimum, first quartile, median, third quartile and maximum. The minimum and maximum are computed excluding extreme values according to the interquartile range

6.2.2 Models robustness

If we focus now our robustness analysis on models using the standard framework first, we see that *std_NS* increases stdev over *std_BASE*, both *std_NO* and *std_LB* slightly decreases it for almost all evaluation measures. Moreover, this behaviour is accentuated in the *std_NO_LB* model.

Looking at the late fusion models, we see that the *late_NO_LB_NS* model decreases stdev over all evaluation measures compared to *late_NO_NS*, dividing it by two for the PRES@100, Recall@100 and MAP@100.

Analyzing the results globally, we see that while *std_NO_LB* yields slightly worse results than *late_NO_NS* and *late_NO_LB_NS*, it is a more robust model as its stdev is lower especially for the MAP(D) and PREC(D) measures.

Table 5 Values of some parameters present in our models. λ (nth) represent the λ parameter in the nth HGAT layer

Model	*std_LB*	*std_NO_LB*	*std_LB_NS*	*late_NO_NS*	*late_NO_LB_NS*
λ (1st)	0.54 ± 0.11	0.38 ± 0.04	0.71 ± 0.17	-	0.41 ± 0.09
λ (2nd)	0.44 ± 0.09	0.23 ± 0.05	0.39 ± 0.16	-	0.27 ± 0.07
λ (3rd)	0.31 ± 0.06	0.18 ± 0.04	0.76 ± 0.07	-	0.87 ± 0.08
α	-	-	-	0.33 ± 0.21	0.31 ± 0.08

Table 6 Relative importance given by the HGAT variant in two models to each relation type in our document graph representation. Values are softmaxed with respect to relation type, averaged between the runs of the models and presentend under percentage form. Green (left columns) represent the *std_BASE* model while orange (right columns) represent the *std_NO_LB* model

Relation	1st layer		2nd layer		3rd layer	
HGAT layer						
order	**10.87**	***8.24***	**5.59**	***5.96***	**9.87**	***4.01***
$order_i$	**22.11**	***6.79***	**6.41**	***4.39***	**13.22**	***3.81***
structural	**3.44**	***0.93***	**2.13**	***3.68***	**31.11**	***6.23***
$structural_i$	**24.12**	***4.37***	**49.42**	***4.83***	**15.14**	***0.82***
internal	**0.11**	***0.86***	**0.04**	***0.39***	**1.78**	***0.20***
$internal_i$	**7.78**	***9.16***	**6.98**	***2.92***	**3.6**	***2.72***
external	**4.51**	***0.05***	**3.74**	***0.31***	**0.08**	***0.02***
$external_i$	**17.18**	***7.08***	**11.28**	***0.48***	**3.32**	***0.17***
SELF	**9.88**	***62.00***	**14.41**	***77.00***	**21.88**	***82.00***

To visualize more accurately the difference in results of these three models, we present boxplots of their performances for each evaluation feature in Fig. 4. If we compare the models using the late framework, we see that the *late_NO_LB_NS* model has a greater first quartile than the *late_NO_NS* model for three evaluation measures (Recall, MAP(D) and PREC(D)), and a greater minimum value for every evaluation features. This shows that, despite having a lower maximum value for three evaluation features, the *late_NO_LB_NS* model is less subject to random initialization than the *late_NO_NS* model.

If we compare the model using the standard framework with the two other models, we see that for the Recall@100 and the measures focused on the passages (MAP(D) and PREC(D)), the minimum value of the *std_NO_LB* model is comparable or greater than the 1st quartile of the *late_NO_NS* and *late_NO_LB_NS* models. Moreover, for all measures except the PRES@100, the minimum value of *std_NO_LB* model is greater than the minimum of the other two. So, despite having worse maximum performances, the *std_NO_LB* model reveals to be competitive by its stability.

Takeaways focusing on our HGAT architecture modifications are that the NO and LB modifications reduce model variability, with a combination of the two increasing stability even more. Even though models built with the late fusion framework give better results, the combination of the NO and LB modifications with the standard framework yields comparable results with increased robustness.

6.2.3 Parameter analysis

We analyse now some parameters of our models in order to better understand their behaviour. We focus on two parts of the models: **(1)** the parameters λ and α controlling

the importance of content and context at different points in the models (highlighted in Table 5), **(2)** the relative importance given to each relation type in the document graph representation (highlighted in Table 6). In the latter, we study the differences between the unmodified *std_BASE* model and one of our most effective model: *std_NO_LB*. To obtain a single value representing the importance of a relation, we average the 180 value of its corresponding weight vector and compute its softmax with the other averaged weight vectors corresponding to the other relations. In Table 6, we rapport these values per HGAT layers and averaged between the several runs of each model. We note that since the λ parameters serve as balance between content and context in the *std_NO_LB* model, we assign $1 - \lambda$ to the SELF weight relation, and multiply every other by λ.

Importance of content and context

Concerning the λ parameter, we see that there is similar behaviour between *std_LB* and *std_NO_LB* showing that context is less taken into account in later layers. Considering how each HGAT layer allows information from nodes one hop further to reach the target node, this shows that the models give more importance to closer contextual nodes. In the *std_LB* model, target node content information can flow to other nodes during one layer and come back during the next under the form of contextual information. This phenomenon is not possible in the *std_NO_LB* model with the NO architecture modification, explaining why it considers content as more important, yielding lower lambda values. For the *std_LB_NS* model, we notice that the model gives much more importance to the context in the 1st and 3rd layer. We hypothesize that due to the NS modification, it learns to send target node information to its neighbouring nodes in the 1st layer and send it back in the 3rd layer. We see that the *late_NO_LB_NS* model behave similarly as the *std_NO_LB* model for the 1st and 2nd layer but prioritize contextual information in the 3rd layer. This is explained by the fact that the HGAT variant it uses only computes representation of the target node's context.

Concerning the α parameter, we see that the *late_NO_NS* and *late_NO_LB_NS* models share a similar mean value (0.33 and 0.31 respectively). However, we see that the later shows more stability (0.21 and 0.08 standard deviation values, respectively), comforting the idea that the LB architecture modification brings stability to the model.

Relative relation weight

Focusing on the *std_BASE* model (green values on the left column for each layer), we see that the model prioritizes contextual information over node content (SELF) in all layers. Moreover, we see that it focuses on inverse relations in the two first layers (24.12% and 22.11% for the $order_i$ and $structural_i$ relations in the 1st layer, respectively) and the "regular" relation in the last one (31.11% for the *structural* relation). Based on the idea that a passage content is central in estimating its relevance, our intuition is that in the first layers the model prioritizes relations from which the target node can send content information to its neighbours. Then in the last layer, the model prioritizes relations from which the target node can get its content information back from the neighbours.

Focusing on the *std_NO_LB* model (orange values for the right column for each layer), we see that the model prioritizes content information over context in all layers. Unlike *std_BASE*, this model can't send target node information to its neighbours which explains why the SELF relation is given such importance (62%, 77% and 82% in the 1st, 2nd and 3rd layer respectively). If we analyze the non-SELF relations, we see that the passage order as well as internal references relations have higher importance than others in the first layer, while the structural relation is privileged in the last layer. A potential explanation is that the model considers that only the target node's direct neighbours are important with respect to

the passage order and references relations, and that nodes further in the graph are important with respect to the structural relations.

6.2.4 Comparison with the state-of-the-art

We provide performance comparison with a State-of-the-art non-neural passage contextualization model ($QSF_{sect.Prop.AVG}$ (Albarede et al. (2021)), two baseline models that do not consider context (fine-tuned ColBERT (Khattab and Zaharia (2020)), BM25 (Robertson et al. (1995))) and an approach that focuses on the patent specific query generation step (Andersson et al. (2016)), namely *Query-gen*. Note that some of our HGAT variants are equivalent to the models presented in (Albarede et al. (2022)), which is why we do not compare with the results given in that paper.

For our models, and due to the nature of our experiments, we only rapport the mean results and are unable to perform statistical significance tests. Furthermore, for clarity reasons, we only rapport our three highest performing models: *std_NO_LB*, *late_NO_LB_NS* and *late_NO_NS*.

We see on Table 7 that our approaches outperform the other state-of-the-art contextualization approach $QSF_{sectionPropagateAVG}$ (up to 16.9%, 49.7% and 14.9% on the PRES, MAP and MAP(D) respectively when compared with *late_NO_NS*) and outperform all non-contextualization approaches (up to 21.1%, 35.2% and 58.2% on the PRES, MAP and MAP(D) respectively when comparing *late_NO_NS* to $Query - gen$) except for the *Query-gen* model on the PREC(D) measure. We hypothesize that this is because the *Query-gen* model retrieves less passages per document: decreasing the number of false positives (higher PREC(D)) but also the number of true positives (lower MAP(D)).

7 Conclusion

In this work, we studied the use of HGATs, complex models that compute contextualized representations of nodes in a graph, to perform Passage Retrieval. To do so, we first designed different HGAT architectures according to some desirable behaviours. We then integrate these HGAT variants into two passage ranking frameworks yielding several passage ranking models capable of dealing with any document graph representation. Experiments on the CLEF-IP2013 Passage Retrieval task show that classical HGATs do not have the most suited architecture to be used for the task, and that some modifications significantly improve the performances and outperform the state-of-the-art by up to 49.7% on the MAP evaluation measure. Moreover, a robustness study shows that judicious modifications brought to HGAT models vastly improve the overall model stability to random parameter initialization. As stated earlier, our approach is not defined for specific documents in mind and can be used for various types of documents having different intra- and inter-links. For future works, it would be interesting to test our models on other types of non-patent documents (e.g. differently structured or unstructured documents).[2]

[2] http://www.ifs.tuwien.ac.at/clef-ip/index.html

Table 7 Comparison of our approaches (three bottom lines) with the state-of-the-art over CLEF-IP2013. We rapport the mean values of our models. Boldface indicates best result

	Model	PRES@100	Recall@100	MAP@100	MAP(D)	PREC(D)
Content only	*BM*25 (Robertson et al. (1995))	0.385	0.482	0.125	0.142	0.21
Content & Context	*ColBERT* (Khattab and Zaharia (2020))	0.402	0.518	0.161	0.145	0.214
	Query-gen (Andersson et al. (2016))	0.444	0.560	0.187	0.146	**0.282**
	$QSF_{sec.Prop.AVG}$ (Albarede et al. (2021))	0.460	0.609	0.169	0.201	0.237
	std_NO_LB	0.522	**0.615**	0.239	0.219	0.263
	late_NO_NS	**0.538**	0.613	**0.253**	**0.231**	0.266
	late_NO_LB_NS	0.534	0.612	0.252	0.227	0.262

A Appendix

We give a complete description of how our HGAT variants compute a node representation.

NO_LB variant

- To compute the intermediate representation of the target node, we use equation (3).
- To compute the intermediate representation of every other node, we use equation (7):

$$h_i' = (1-\lambda) * MultiHead_{\phi(e_{ii})}(h_i, h_i, h_i) + \lambda * \Big(\sum_{j \in N_i, j \neq i, j \neq tgt} \sum_{e' \in e_{ij}} softmax(W_{\phi(e')}) * MultiHead_{\phi(e')}(h_i, h_j, h_j) \Big) \quad (7)$$

LB_NS variant

- To compute the intermediate representation of the target node, we use equation (8):

$$h_i' = \sum_{j \in N_i, j \neq tgt} \sum_{e' \in e_{ij}} softmax(W_{\phi(e')}) * MultiHead_{\phi(e')}(h_i, h_j, h_j) \quad (8)$$

- To compute the intermediate representation of every other node, we use equation (3).

NO_NS variant

- To compute the intermediate representation of the target node, we use equation (8).
- To compute the intermediate representation of every other node, we use equation (2).

NO_LB_NS variant

- To compute the intermediate representation of the target node, we use equation (8).
- To compute the intermediate representation of every other node, we use equation (7).

Author contributions All authors contributed to the study conception and design. The first draft of the manuscript was written by Lucas Albarede and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Availability of data and materials The code used in this work is property of Schneider Electric. The CLEF-IP2013 dataset is public and can be accessed at this website.

Declarations

Ethical Approval Not applicable in the context of this work.

Conflict of interest Lucas Albarede declares support from the Schneider Electric company and from the french Association Nationale de la Recherche et de la Technologie for the submitted work. Claude Le Pape-Gardeux, Sylvain Marié and Trinidad Chardin-Segui declare support from the Schneider Electric company for the submitted work. Philippe Mulhem declares support from the french National Center for Scientific Research for the submitted work. Lorraine Goeuriot declares supports from the Grenoble Alpes Université for the submitted work. All authors declare no financial relationships with any organisations that might have

an interest in the submitted work in the previous three years; no other relationships or activities that could appear to have influenced the submitted work.

Funding This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003), as well as the Association Nationale de la Recherche et de la Technologie (ANRT).

References

Albarede, L., Mulhem, P., Goeuriot, L., Le Pape-Gardeux, C., Marie, S. and Chardin-Segui, T. (2021). Passage retrieval in context: Experiments on patents. Proceedings of CORIA'21. Proceedings of coria'21. Grenoble, France. https://hal.archives-ouvertes.fr/hal-03230421

Albarede, L., Mulhem, P., Goeuriot, L., Le Pape-Gardeux, C., Marié, S. and Chardin-Segui, T. (2022). Passage Retrieval on Structured Documents using Graph Attention Networks. Proceedings of ECIR 2022. Proceedings of ecir 2022. Stavanger, Norway. https://hal.archives-ouvertes.fr/hal-03626054

Andersson, L., Lupu, M., Palotti, J.a., Hanbury, A. & Rauber, A. (2016). When is the time ripe for natural language processing for patent passage retrieval? Proceedings of the 25th ACM International on Conference on Information and Knowledge Management Proceedings of the 25th acm international on conference on information and knowledge management (p. 1453-1462). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/2983323.2983858

Arnold, S., van Aken, B., Grundmann, P., Gers, F.A. and Löser, A. (2020). Learning contextualized document representations for healthcare answer retrieval. CoRR abs/2002.00835) arXiv.org/abs/2002.00835

Arvola, P., Junkkari, M. and Kekäläinen, J. (2005). Generalized contextualization method for xml information retrieval. Proceedings of the 14th ACM International Conference on Information and Knowledge Management Proceedings of the 14th acm international conference on information and knowledge management (p. 20-27). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/1099554.1099561

Arvola, P., Kekäläinen, J. & Junkkari, M. (2008). The effect of contextualization at different granularity levels in content-oriented xml retrieval. Proceedings of the 17th ACM Conference on Information and Knowledge Management Proceedings of the 17th acm conference on information and knowledge management (p. 1491-1492). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/1458082.1458350

Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv. arxiv:1409.0473

Beigbeder, M. (2010). Focused retrieval with proximity scoring. Proceedings of the 2010 ACM Symposium on Applied Computing Proceedings of the 2010 acm symposium on applied computing (p. 1755-1759). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/1774088.1774462

Bendersky, M. & Kurland, O. (2008). Utilizing passage-based language models for document retrieval. Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval Proceedings of the ir research, 30th european conference on advances in information retrieval (p. 162-174). Berlin, HeidelbergSpringer-Verlag.

Callan, J.P. (1994). Passage-level evidence in document retrieval. Proceedings of the 17th annual international acm sigir conference on research and development in information retrieval (p. 302-310). Berlin, HeidelbergSpringer-Verlag.

Carmel, D., Shtok, A. & Kurland, O. (2013). Position-based contextualization for passage retrieval. In: Proceedings of the 22nd acm international conference on information & knowledge management (p. 1241-1244). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/2505515.2507865

Chen, L., Li, J., Gong, Z., Zhang, M. & Zhou, G. (2022). One type context is not enough: Global context-aware neural machine translation. ACM Transactions on Asian and Low-Resource Language Information Processing. https://doi.org/10.1145/3526215 (Just Accepted)

Cohen, D., Jordan, S.M. & Croft, W.B. (2019) Learning a better negative sampling policy with deep neural networks for search. Proceedings of the 2019 acm sigir international conference on theory of information retrieval (p. 19-26). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/3341981.3344220

Cui, H., Lu, J., Ge, Y. & Yang, C. (2022) How can graph neural networks help document retrieval: A case study on cord19 with concept map generation. arXiv. arxiv:2201.04672

Fernández, R., Losada, D. & Azzopardi, L. (2011). Extending the language modeling framework for sentence retrieval to include local context. Inf. Retr. 14, 355-389. https://doi.org/10.1007/s10791-010-9146-4

Gobeill, J. & Ruch, P. (2012). Bitem site report for the claims to passage task in CLEF-IP 2012. P. Forner, J. Karlgren and C. Womser-Hacker (Eds.), CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012 CLEF 2012 evaluation labs and workshop, online working notes, rome, italy, september 17-20, 2012 (Vol. 1178). CEUR-WS.org. http://ceur-ws.org/Vol-1178/CLEF2012wn-CLEFIP-GobeillEt2012.pdf

Hao, X., Zhou, Y., Wu, D., Zhang, W., Li, B. & Wang, W. (2021). Multi-feature graph attention network for cross-modal video-text retrieval. Proceedings of the 2021 international conference on multimedia retrieval (p. 135-143). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/3460426.3463608

Hofstätter, S., Mitra, B., Zamani, H., Craswell, N. and Hanbury, A. (2021). Intra-document cascading: Learning to select passages for neural document ranking. arXiv. arxiv:2105.09816

Hong, W., Zhang, Z., Wang, J. and Zhao, H. (2022). Sentence-aware contrastive learning for open-domain passage retrieval. Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers) (p.S 1062–1074). Dublin, IrelandAssociation for Computational Linguistics. https://aclanthology.org/2022.acl-long.76

Kaszkiel, M., Zobel, J. & Sacks-Davis, R. (1999). Efficient passage ranking for document databases. ACM Transactions on Information Systems (TOIS) 17(4), 406-439. https://doi.org/10.1145/326440.326445

Kekäläinen, J., Arvola, P. & Junkkari, M. (2018). Contextualization in structured text retrieval. In L. Liu & M.T. Ozsu (Eds.),Encyclopedia of Database Systems Encyclopedia of database systems (p.S 611–613). New York, NYSpringer New York. https://doi.org/10.1007/978-1-4614-8265-9_81

Khattab, O. & Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over BERT. CoRR, abs/2004.12832) arxiv:2004.12832

Kingma, D.P. & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. arxiv:1412.6980

Krikon, E., Kurland, O., & Bendersky, M. (2011). Utilizing inter-passage and inter-document similarities for reranking search results. *ACM Transactions on Information Systems, 29*(1), 1–28. https://doi.org/10.1145/1877766.1877769

Li, X., de Rijke, M., Liu, Y., Mao, J., Ma, W., Zhang, M. and Ma, S. (2020). Learning better representations for neural information retrieval with graph information. Proceedings of the 29th acm international conference on information & knowledge management (p. 795-804). New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3340531.3411957

Liu, J., Liu, J., Yang, Y., Wang, J., Wu, W., Zhao, D. and Yan, R. (2022). Gnn-encoder: Learning a dual-encoder architecture via graph neural networks for passage retrieval. arXiv. arxiv:2204.08241

Liu, Y., Hashimoto, K., Zhou, Y., Yavuz, S., Xiong, C. and Yu, P.S. (2021a). Dense hierarchical retrieval for open-domain question answering. arXiv. arxiv:2110.15439

Liu, Y., Hashimoto, K., Zhou, Y., Yavuz, S., Xiong, C. and Yu, P.S. (2021b). Dense hierarchical retrieval for open-domain question answering. arXiv. arxiv:2110.15439

Lu, J., Ábrego, G.H., Ma, J., Ni, J. and Yang, Y. (2020). Neural passage retrieval with improved negative contrast. CoRR, abs/2010.12523 arxiv:2010.12523

Lu, J., Hernandez Abrego, G., Ma, J., Ni, J. and Yang, Y. (2021). Multi-stage training with improved negative contrast for neural passage retrieval. Proceedings of the 2021 conference on empirical methods in natural language processing (p.S 6091–6103). Online and Punta Cana, Dominican Republic Association for Computational Linguistics. https://aclanthology.org/2021.emnlp-main.492 https://doi.org/10.18653/v1/2021.emnlp-main.492

Macdonald, C. & Tonellotto, N. (2020). Declarative experimentation in information retrieval using PyTerrier. Proceedings of the 2020 ACM SIGIR on international conference on theory of information retrieval. ACM. https://doi.org/10.1145/3409256.3409829

Macdonald, C., Tonellotto, N. and Ounis, I. (2021). On single and multiple representations in dense passage retrieval. CoRR, abs/2108.06279) arxiv:2108.06279

Mahdabi, P., Gerani, S., Huang, J.X. and Crestani, F. (2013). Leveraging conceptual lexicon: Query disambiguation using proximity information for patent retrieval. Proceedings of the 36th international acm sigir conference on research and development in information retrieval (p. 113-122). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/2484028.2484056

Mahdabi, P., Keikha, M., Gerani, S., Landoni, M. and Crestani, F. (2011). Building queries for prior-art search. A. Hanbury, A. Rauber and A.P. de Vries (Eds.), Multidisciplinary information retrieval (p.S 3–15). Berlin, HeidelbergSpringer Berlin Heidelberg.

Mass, Y. & Mandelbrod, M. (2005). Component ranking and automatic query refinement for xml retrieval. N. Fuhr, M. Lalmas and S. Malik (Eds.), Advances in xml information retrieval (p.S 73–84). Berlin, HeidelbergSpringer Berlin Heidelberg.

Murdock, V. & Croft, W.B. (2005). A translation model for sentence retrieval. Proceedings of human language technology conference and conference on empirical methods in natural language processing (p.S 684–691). Vancouver, British Columbia, CanadaAssociation for Computational Linguistics. https://www.aclweb.org/anthology/H05-1086

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R. and Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. CoRR, abs/1611.09268) http://arxiv.org/abs/1611.09268

Nogueira, R.F. & Cho, K. (2019). Passage re-ranking with BERT. CoRR, abs/1901.04085) arxiv:1901.04085

Norozi, M.A. & Arvola, P. (2013). Kinship contextualization: Utilizing the preceding and following structural elements. Proceedings of the 36th international acm sigir conference on research and development in information retrieval (p. 837-840). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/2484028.2484111

Norozi, M.A., Arvola, P. and de Vries, A.P. (2012). Contextualization using hyperlinks and internal hierarchical structure of wikipedia documents. Proceedings of the 21st acm international conference on information and knowledge management (p. 734-743). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/2396761.2396855

Norozi, M.A., de Vries, A. and Arvola, P. (2012). Contextualization from the Bibliographic Structure. Proceeding of the ecir 2012 workshop on task-based and aggregated search (tbas2012), page 9.

Ogilvie, P. & Callan, J. (2005). Hierarchical language models for xml component retrieval. N. Fuhr, M. Lalmas and S. Malik (Eds.), Advances in xml information retrieval (p.S 224–237). Berlin, HeidelbergSpringer Berlin Heidelberg.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox and R. Garnett (Eds.), Advances in neural information processing systems 32 (p.S 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Piroi, F., Lupu, M. and Hanbury, A. (2013). Overview of clef-ip 2013 lab. P. Forner, H. Müller, R. Paredes, P. Rosso and B. Stein (Eds.), Information access evaluation. multilinguality, multimodality, and visualization (p.S 232–249). Berlin, HeidelbergSpringer Berlin Heidelberg.

Qi, Y., Zhang, J., Liu, Y., Xu, W. and Guo, J. (2020). Cgtr: Convolution graph topology representation for document ranking. Proceedings of the 29th ACM International Conference on Information & Knowledge Management (p. 2173-2176). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/3340531.3412073

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M. and Gatford, M. (1995). Okapi at trec-3. Overview of the third text retrieval conference (trec-3) (Overview of the Third Text REtrieval Conference (TREC-3) ed., p. 109-126). Gaithersburg, MD: NIST. https://www.microsoft.com/en-us/research/publication/okapi-at-trec-3/

Santhanam, K., Khattab, O., Potts, C. and Zaharia, M.A. (2022). Plaid: An efficient engine for late interaction retrieval. ArXiv:abs/2205.09707

Sheetrit, E., Shtok, A. and Kurland, O. (2019). A passage-based approach to learning to rank documents. ArXiv:abs/1906.02083

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. and Polosukhin, I. (2017). Attention is all you need. arXiv. arxiv:1706.03762

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. and Bengio, Y. (2017). Graph attention networks. arXiv. arxiv:1710.10903

Wang, X., Ji, H., Shi, C., Wang, B., Cui, P., Yu, P.S. and Ye, Y. (2019). Heterogeneous graph attention network. CoRR. arXiv:1903.07293

Wu, N., Liang, Y., Ren, H., Shou, L., Duan, N., Gong, M. and Jiang, D. (2022). Unsupervised context aware sentence representation pretraining for multi-lingual dense retrieval.

Xie, Q., Huang, J., Du, P., Peng, M. and Nie, J.-Y. (2021). Graph topic neural network for document representation. Proceedings of the web conference 2021 (p. 3055-3065). New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3442381.3450045

Xu, P., Chen, X., Ma, X., Huang, Z. and Xiang, B. (2021). Contrastive document representation learning with graph attention networks. arXiv. arxiv:2110.10778

Xue, X. & Croft, W.B. (2009). Automatic query generation for patent search. Proceedings of the 18th acm conference on information and knowledge management (p. 2037-2040). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/1645953.1646295

Zhang, T., Liu, B., Niu, D., Lai, K. and Xu, Y. (2018). Multiresolution graph attention networks for relevance matching. Proceedings of the 27th ACM International Conference on Information and Knowledge Management) https://doi.org/10.1145/3269206.3271806

Zhang, Y., Liu, C., Luo, A., Xue, H., Shan, X., Luo, Y. and Wang, H. (2021). Mira:leveraging multi-intention co-click information in web-scale document retrieval using deep neural networks. Proceedings of the Web Conference 2021 Proceedings of the web conference 2021 (p. 227-238). New York, NY, USAAssociation for Computing Machinery. https://doi.org/10.1145/3442381.3449865

Zhao, S., Su, C., Sboner, A. & Wang, F. (2019). GRAPHENE. Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Proceedings of the 28th ACM international conference on information and knowledge management. ACM. https://doi.org/10.1145/3357384.3358038