Constrained Information Retrieval for Long-Tail Knowledge Extraction

Nicolas Lazzari[1,2,*,†], Arianna Graciotti[1,†] and Valentina Presutti[1]

[1]LILEC, University of Bologna, Via Cartoleria, 5, Bologna 40124, Italy
[2]Computer Science Department, University of Pisa, Largo B. Pontecorvo, 3, Pisa 56127, Italy

Abstract

Information retrieval is a critical step in frameworks that extract structured knowledge from unstructured text. It is essential for NLP tasks such as open domain question answering, entity linking, and relation extraction. Modern retrieval frameworks often rely on a retriever component, typically based on a bi-encoder architecture. The bi-encoder encodes the input text and the knowledge base, calculating dot product similarities to find relevant candidates. Bi-encoders are usually based on pre-trained language models or learned text embedding models. Such models rely heavily on the training data and perform sub-optimally in domain-specific tasks or scenarios involving unpopular entities and long-tail relations. This is known as popularity bias. We propose a method that leverages explicit knowledge from curated knowledge graphs, such as Wikidata, to improve retrieval performance by filtering implausible candidates. Plausibility is defined through Answer Set Programming and is independent of the retriever. We show that it consistently improves the accuracy of the retrieval system on less popular entities by evaluating benchmarks of historical documents.

Keywords

Long-tail Knowledge Extraction, Information Retrieval, Entity Linking, KGs, LLMs

1. Introduction

Information retrieval (IR) is a fundamental task in Artificial Intelligence, with applications in Natural Language Processing (NLP) and image processing. It is used to extract structured knowledge from unstructured text or train complex classifiers on unsupervised data. Recently, interest in IR has grown due to its role in Retrieval Augmented Generation (RAG) methods [1], which can significantly enhance different aspects of Large Language Models.

Modern IR methods combine two neural network architectures: a *bi-encoder* and a *cross-encoder*[2]. The bi-encoder encodes documents into dense vector representations that reflect document similarity [3]. It is used to retrieve relevant documents from a Knowledge Base (KB) based on the vector representations of both the input and the documents in the KB. Relevance is defined as a function of the similarity between two vectors, with similarity and distance metrics serving as proxies for relevance. The cross-encoder then takes the candidates retrieved by the bi-encoder and ranks them according to the task at hand. For example, in Question Answering (QA), an input query (the question) is used to retrieve relevant documents from the KB. The bi-encoder thus acts as a filtering process on the KB, allowing the cross-encoder to rank only a subset of the entire KB.

Although similarity and distance measures between vectors have yielded impressive results, they often suffer a significant drawback: *popularity bias* [4]. This refers to the tendency of these measures to favour frequently occurring documents while underperforming on less common ones. These less common documents, along with the entities and relations they contain, are typically referred to as *long-tail* knowledge.

EKAW 2024: EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), November 26-28, 2024, Amsterdam, The Netherlands.

[*]Corresponding author.

[†]These authors contributed equally.

nicolas.lazzari3@unibo.it (N. Lazzari); arianna.graciotti@unibo.it (A. Graciotti); valentina.presutti@unibo.it (V. Presutti)
0000-0002-1601-7689 (N. Lazzari); 0009-0004-7918-809X (A. Graciotti); 0000-0002-9421-8566 (V. Presutti)

Popularity bias has been widely studied from diverse perspectives, including its impact on LLMs' zero- and few-shot learning [5, 6, 7], training data memorization [8, 9], and privacy concerns [10]. It is also linked to other types of bias at the level of training data and model predictions, such as gender and origin biases [11, 12]. For instance, popularity bias has been observed to undermine the performance of popular LLMs (e.g. ChatGPT [13]) on domain-specific tasks, such as QA on historical named entities related to music, particularly when questions involve women's Wikipedia biographies [14].

Popularity bias is a direct consequence of the bi-encoder architecture. Bi-encoders are heavily dependent on their training data, often used in an unsupervised fashion, and are therefore prone to over-representing popular documents while neglecting less common ones. If the bi-encoder is biased toward popular documents, less popular documents have a reduced chance of being analyzed by the cross-encoder. This becomes particularly problematic in tasks that require the bi-encoder to prioritize recall over precision, such as Entity Linking (EL) or domain-specific QA.

Our approach aims to mitigate these issues by enforcing *logical plausibility* in the bi-encoder model. Instead of comparing an input document to the entire KB, we only consider documents that meet a set of logical constraints grounded in trusted Knowledge Graphs (KGs), such as Wikidata [15]. We define the logical plausibility of a document using Answer Set Programming (ASP), a logic programming technique that relies on stable model semantics and specialized solvers to handle large amounts of data and constraints. Unlike similar approaches, our method is independent of the bi-encoder and cross-encoder architectures and requires no additional model training.

We experiment with several datasets of historical documents annotated for the EL task. Historical documents are known for containing long-tail entities, and SotA entity linkers are mainly trained on contemporary datasets, largely extrapolated from the internet [16, 17]. This enables us to evaluate our approach on benchmarks composed of documents affected by popularity bias in a task highly sensitive to the bi-encoder's recall. We assess the performance using IR measures and by analyzing the results of the retrieval process. Our findings demonstrate that our approach significantly improves the recall of general-purpose bi-encoders, regardless of their underlying architecture, outperforming specialized models. Our contribution can be summarized as the proposal of a general method based on ASP that enhances the recall of an IR system by applying logical plausibility constraints.

The rest of the paper is organized as follows: in Section 2, we review similar approaches that enforce logical constraints in IR systems and popular bi-encoder-cross encoder architectures for EL. In Section 3 we provide an informal introduction to ASP. In Section 4 we describe our method, and we experiment with it in Section 5. Finally, in Section 6, we discuss our results and in Section 7, we summarize our work and highlight future works.

2. Related works

In this section, we review the most representative recent works in QA and EL, since they are the ones mostly influenced by the popularity bias in bi-encoders, by highlighting how they engage with the problem of long-tail knowledge and the role that the retrieval module of their architectures plays in addressing such a problem.

In the realm of QA tasks, Kandpal et al. [4] demonstrate a strong correlation between the knowledge acquired by LLMs and the frequency of that information in their pre-training datasets. LLMs perform better at answering questions when the required information appears frequently in the pre-training data. In particular, increasing the model parameters improves knowledge retention while greatly increasing the effort required to train and maintain the model. RAG systems are a promising approach to mitigate these limitations. Nonetheless, even when relying on simple retrievers, their effectiveness is sensible to the distribution of relevant documents [18].

Indeed, Mallen et al. [19] showed that increasing model size provides limited benefits for less frequent information. The study demonstrates that RAG can improve LM performance on long-tail data but may introduce errors for popular entities.

Similarly, Sun et al. [20] demonstrate a consistent decline in LLMs' QA performance from *head* (highly popular) to *tail* (unpopular) entities, regardless of the LLM power. This poses significant issues since entities' popularity is a biased phenomenon. For instance, it has been shown that on tail entities, LLMs have higher performances when answering questions on men rather than on women [14].

Another downstream task heavily impacted by retrieval performance is EL. SotA entity linkers frequently adopt the *retriever-reader* paradigm, where retrieval quality directly affects the linking process. As a result, EL suffers from the challenges of long-tail knowledge and popularity bias. The difficulty in retrieving less popular entities can lead to errors in linking, particularly when the target entities are underrepresented in the KB.

For instance, BLINK [21], a widely-used retrieval-based entity linker that leverages Pre-trained Language Models (PLMs), has been observed to perform unsatisfactorily on long-tail entities mentioned in historical documents [22].

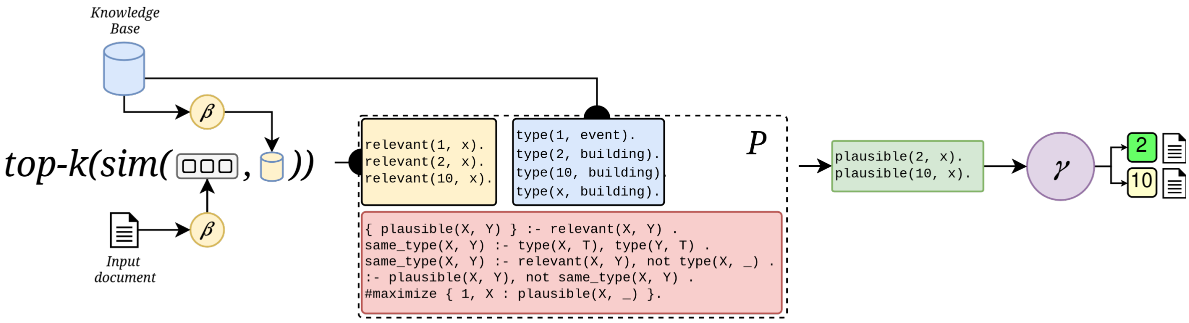
Recent developments in EL brought to its reformulation as an inverse open-domain QA task. An unknown number of questions (corresponding to candidate entities) are retrieved first based on the input documents. Then the model predicts which portions of the text answers (should be linked to) the retrieved questions. EntQA [23] and ReLiK [24] follow this approach, resulting in state-of-the-art results. Despite a different formulation with respect to traditional retrieval-based models, they still suffer from the issues induced by popularity bias in the retrieval phase.

Similarly to our approach, exploiting contextual information to reduce the candidates considered by an IR method has been explored in the past. In the context of EL, Tedeschi et al. [25] leverage NERC information to enhance the EL process. They enrich entity representations with NER information and improve candidate selection by using NER to filter out unlikely candidates both during the training and inference phases. Other similar approaches that exploit type information have been proposed, including reasoning on large KB [26] or including Knowledge Graph information when training the bi-encoder [27]. However, different to our approach, all of these approaches require a dedicated training procedure and are often architecture dependant. By relying on an expressive logic programming paradigm independent of the KB, our approach applies to any bi-encoder architecture and any KB. Moreover, it is possible to enforce logical constraints that are tightly dependent on the dataset at hand.

3. Answer Set Programming

Answer Set Programming (ASP) [28] is a logic programming technique that aims at solving problems in a declarative fashion. The programmer specifies a set of *rules* that characterize the problem in the form of logical constraints and a set of *facts* encoding some data in the form of a KB. Using a solver and automated reasoning techniques, new information is derived based on the asserted facts. Unlike Prolog, ASP is based on the concept of answer sets. Informally, an answer set is a set of facts that can be inferred from the asserted ones using the provided rules while maintaining logical consistency with those rules. In their most general form, ASP programs are NP-complete, meaning that finding the answer sets to a program is not always computationally feasible. Nonetheless, efficient solvers handling a large number of rules and facts have been developed (e.g. clingo [29]). Additionally, modern solvers support advanced features that are generally not available in Prolog, such as preferences over different answer sets, the use of specialized solvers for numerical constraints, as well as probabilistic [30] and neuro-symbolic [31] extensions. Despite a high expressivity, defining ASP programs is intuitive and can be done through the use of a Controlled Language [32], allowing domain experts to directly formalize logic plausibility constraints.

In this work, we generate facts from the use of the bi-encoder and manually assert rules based on the domain of the application. The answer set of a program is, hence, the set of candidates that have a high similarity with an input document and are logically consistent with the constraints imposed by the domain.



```
relevant(1, x).
relevant(2, x).
relevant(10, x).
```

```
type(1, event).
type(2, building).
type(10, building).
type(x, building).
```

```
{ plausible(X, Y) } :- relevant(X, Y) .
same_type(X, Y) :- type(X, T), type(Y, T) .
same_type(X, Y) :- relevant(X, Y), not type(X, _) .
:- plausible(X, Y), not same_type(X, Y) .
#maximize { 1, X : plausible(X, _) }.
```

```
plausible(2, x).
plausible(10, x).
```

Figure 1: Visual representation of our method. The reference Knowledge Base and the input documents are both encoded using the bi-encoder β. The most relevant documents are then extracted and used to construct the program P. Note that the rules in P are formalized such that a candidate is considered plausible if its type is not asserted. Finally, the output of the program P is fed to the cross-encoder γ.

4. Methodology

Without loss of generality, we will refer to a general bi-encoder model as the function β that takes as input a document and outputs a vector $v \in \mathbb{R}^n$ and to a general cross-encoder model as the function γ that takes as input two vectors $\mathbf{x}$ and $\mathbf{y}$ computed using β and outputs a score that approximates the relevance of the document $\mathbf{y} \in [0, 1]$ given the input document $\mathbf{x}$.

The functions β and γ can be implemented using different techniques, including pre-trained language models, recurrent neural networks, or transformer models. Their composition results in the IR model f.

Our approach consists of a filtering process between β and γ using the ASP program P, shown in Figure 1. The program P is composed of two main components: the facts related to the input document and to the retrieved candidates and the plausibility constraints.

The plausibility constraints are domain-dependant while the facts depend on the output of the bi-encoder and the requirement posed by the plausibility constraints. The facts are extracted from a reference KB, such as a Knowledge Graph. In this work, we rely on the metadata and annotations provided by the dataset alongside information retrieved from Wikidata. To avoid overloading the program P we only consider facts related to the most relevant documents found by the bi-encoder. Even though ASP solvers can handle large quantities of data, this minimizes the impact of the filtering process in the IR system.

The program P first enumerates the relevant entities found by the bi-encoder and their data using binary predicates. The most important predicate is `relevant/2`, which asserts that a document is relevant for another document. Depending on the information available, other assertions can be added to P, such as `type/2` to assert the type of the content described in the document, `year/2` for the year of publication of the document and so on. Figure 1 shows an example of facts assertions of P.

Secondly, the constraints for logical plausibility are asserted. The program P follows a generate-and-test approach to ASP programming. Informally, the solver first *generates* an answer set where all relevant documents are considered plausible, using the predicate `plausible/2`. It then removes from the answer set all the `plausible/2` assertions that are not consistent with the constraints. In the example of Figure 1, the program P removes all the relevant documents whose type does not match the input document's type.

5. Experiments

In this section, we apply the methodology described in Section 4 to several datasets containing historical documents with NERC and EL annotations. Historical documents often include long-tail entities making them a suitable testing ground for demonstrating to what extent applying logical plausibility constraints to bi-encoder architectures can improve the performance of these IR modules on long-tail knowledge.

```
% Generate plausible candidates
{ plausible(X, Y) } :- relevant(X, Y) .
% Define type-plausibility and remove implausible candidates
same_type(X, Y) :- type(X, T), type(Y, T) .
same_type(X, Y) :- relevant(X, Y), not type(X, _) .
:- plausible(X, Y), not same_type(X, Y) .
% Define year-plausibility and remove implausible candidates
compatible_year(X, Y) :- year(X, YX), year(Y, YY), YX <= YY .
compatible_year(X, Y) :- relevant(X, Y), not year(X, _) .
:- plausible(X, Y), not compatible_year(X, Y) .
% Compute the answer set with the highest number of plausible candidates
#maximize { 1, X : plausible(X, _) }.
```

Listing 1: ASP program P used to filter implausible candidates.

We experiment with HIPE-2020 (Section 5.1), MHERCL (Section 5.2), AjMC (Section 5.3) and TopRes19th (Section 5.4) using different bi-encoders to assess the method's effectiveness. We ignore documents whose entity is labelled as `NIL`.

We evaluate the output by computing the recall on subsets of different lengths. An output is considered correct if the target entity is within that subset. This metric does not evaluate the final link but rather measures how reliable the results of the bi-encoder are. A high recall on a subset means that the cross-encoder can re-rank the results to perform EL. Clearly, this is impossible if the target document is not within the ones retrieved. In other words, we measure the ability of our method to filter out irrelevant entities such that unpopular but plausible ones emerge.

Moreover, we qualitatively evaluate the results by comparing them to the documents retrieved by ReLiK. For each dataset, we report an example in which ReLiK is able to retrieve the correct candidate while our filtering method fails and vice versa.

KB specifications. We use the KILT KB [33] as our reference to construct the entity index, which includes 5.9 million entities and serves as the foundation for several other retrieval systems [21, 23, 24]. Each entity's textual representation combines its title and opening text from Wikipedia.

We extract additional information from Wikidata for each entity in KILT, focusing on temporal and type information[1].

We manually map the entity types from Wikidata to the NERC information available in each dataset[2]. This mapping ensures that logical plausibility filters tailored to each dataset can be crafted and applied during the EL process.

Filters. For all the datasets, we implement the logical plausibility filters by relying on an ASP program P that leverages time and type information. Namely, we filter out implausible candidates from a type and a time perspective. A candidate is considered *type-plausible* if the entity it describes matches the classification of the named entity at hand, while it is considered *year-plausible* if the date of the entity precedes the one of the entity at hand. For example, if the gold NERC information given in a dataset is a location, we consider implausible all those candidates classified as persons in Wikidata. Similarly, if the entity at hand is mentioned in a periodical issued on a given date, we consider implausible a candidate whose date of birth, or inception date, happened later than that date. If type or (respectively

[1] We use property `P31` for type information. We use various time-related properties, ranging from the highly specific `P569` (`date of birth`) to the more generic `P585` (`point in time`) for time information

[2] The mapping is done according to a pre-defined taxonomy of named entity types available in each dataset. For example, the Wikidata type `Q5` (human) is mapped to the NERC type `person` in MHERCL and HIPE-2020 datasets. The Wikidata type `Q747074` (`commune of Italy`) is mapped to the NERC types `'city', 'location'` in MHERCL, and `loc` in HIPE-2020, etc.

year) information is not available, it is considered type-plausible (year-plausible). The program P is reported in Listing 1.

Sentence Embeddings for entity linking. We rely on sentence embeddings [34] to implement the function β, namely on MPNet[3] [35], a distilled version of RoBERTa[4] [36] and MiniLM[5] [37]. Since these models are not explicitly trained for the EL task, they retrieve multiple documents when a sentence contains multiple named entities. To overcome this limitation, we compute two distinct vectors for the sentence and the mention and linearly project the sentence embedding onto the direction of the mention embedding. Formally, given a sentence s and a mention m we compute the dense vector $\mathbf{v}$ as a scaled version of the vector $\beta(m)$ where the scale factor is computed as the ratio between dot product $\langle \beta(s), \beta(m) \rangle$ and the dot product $\langle \beta(m), \beta(m) \rangle$

$$\mathbf{v} = \frac{\beta(s) \cdot \beta(m)}{\beta(m) \cdot \beta(m)} \beta(m). \tag{1}$$

This allows us to retain the generality of the embedding model while obtaining a vector representation better suited to retrieve documents related to a mention. We compare our results with a retriever model specifically trained for entity linking, ReLiK. Since its retriever does not include the use of an explicit mention, we craft its input using the template `mention: <m> sentence: <s>` as input where `<s>` is the sentence of a document and `<m>` the mention to be linked. We retrieve a total of 300 candidates using both general bi-encoders and ReLiK.

5.1. HIPE-2020

Model		R@10	R@30	R@50	R@100	R@200	R@300
ReLiK [24]		**0.81**	0.90	0.93	0.96	0.97	1.00
MPNet [35]		0.42	0.62	0.73	0.89	0.99	1.00
	+ ASP	0.65	**0.91**	**0.96**	**0.99**	0.99	1.00
distill-RoBERTa [36]		0.39	0.58	0.71	0.83	0.94	1.00
	+ ASP	0.59	0.87	0.94	**0.99**	**1.00**	1.00
MiniLM [37]		0.31	0.49	0.59	0.82	0.96	1.00
	+ ASP	0.51	0.84	0.95	**0.99**	**1.00**	1.00

Table 1
Recall at different levels computed using different bi-encoders on HIPE-2020 dataset. Best results for each model are represented in bold.

The HIPE-2020 dataset [38, 39] comprises historical newspaper articles and classic commentaries in 3 different languages (French, German, and English) published in the 19th and 20th centuries. In this work, we rely on the English dataset (specifically, on the test set[6], which has been designed to evaluate EL methods in the domain of historical knowledge and whose annotations include NERC.

Table 1 reports the results obtained. ReLiK performs best when only 10 candidates are taken into account. However, it performs worse when compared to simpler bi-encoders with the filtering process, which can extract all the relevant entities within the first 200 candidates, with most already retrieved within the first 100 candidates.

Table 2 analyses two example errors in a comparative overview. In the first sentence, only ReLiK retrieves the target candidate. Altough most of the candidates retrieved by MPNet are related to Europe, they do not refer to the correct entity. Even though it is difficult to interpret why the similarity of those

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[4] https://huggingface.co/sentence-transformers/all-distilroberta-v1

[5] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased

[6] HIPE-2020 English test set is available for download at https://github.com/hipe-eval/HIPE-2022-data/blob/main/data/v2.1/hipe2020/en/HIPE-2022-v2.1-hipe2020-test-en.tsv

HIPE-2020 newspaper, 1890 In England and other parts of Europe (*Q21*), horseshoes are now in use, made of cowhide instead of iron.	
Model	Top 10
ReLiK	**Europe [Q46]**, England [Q21], Europe (band) [Q185144], Cowhide [Q12492880], ...
MPNet	Somewhere In Europe [Q18230653], Saint-Setiers [Q625213], List of paramilitary groups [Q25344947], In Europe [Q6009397], Europe Today [Q56222917], ...
MPNet + ASP	Saint-Setiers [Q625213], Padiyam [Q7123802], [Q1064023], Southern Europe [Q27449], European Regions [Q6470668], Geography of the European Union [Q941769], ...
HIPE-2020 newspaper, 1910 Why does Great Britian (*Q23666*) buy its oatmeal of us ?	
Model	Top 10
ReLiK	Great Briton Award [Q5598898], Great British Chefs [Q5598892], Greatest Britons [Q5600941], Britons [Q842438], British national identity [Q3402148], ...
MPNet	Great Britain (disambiguation) [Q294011], Britain and Ireland [Q5598828], British countries [Q4971318], **Great Britain [Q23666]**, ...
MPNet + ASP	**Great Britain [Q23666]**, Britain (place name) [Q3240725], Great Britain at the Hopman Cup [Q5598861], Britiande [Q64441], ...

Table 2
Examples of errors on HIPE-2020 dataset. The mention to be linked is underlined in the sentence, while the correct candidate is highlighted in bold. Candidates are displayed in order by the bi-encoder similarity.

documents is higher when compared to the correct document, one possible reason might lie in the use of the linear projection described in Section 1. Sentence embeddings are not directly optimized for this use, and while we observed good results, there might be cases for which this approach outputs are not well defined. Nonetheless, MPNet can retrieve the correct candidate within the 300 retrieved documents, as can be inferred from Table 1.

In the second sentence, ReLiK retrieves type implausible candidates, such as the organization `Q5598898` (`Great Briton Award`) and the work of art `Q5598899` (`Great British Menu`). MPNet also retrieves implausible candidates, but they are effectively filtered out through the program *P*.

5.2. MHERCL

Model		R@10	R@30	R@50	R@100	R@200	R@300
ReLiK [24]		**0.84**	0.91	0.93	0.96	0.99	1.00
MPNet [35]		0.38	0.65	0.73	0.88	0.97	1.00
	+ ASP	0.72	**0.92**	**0.96**	**0.99**	**1.00**	1.00
distill-RoBERTa [36]		0.39	0.58	0.71	0.82	0.96	1.00
	+ ASP	0.68	0.87	**0.96**	**0.99**	**1.00**	1.00
MiniLM [37]		0.27	0.49	0.61	0.78	0.92	1.00
	+ ASP	0.68	0.89	0.95	**0.99**	**1.00**	1.00

Table 3
Recall at different levels computed using different bi-encoders on MHERCL dataset. Best results for each model are represented in bold.

The Musical Heritage Historical named Entities Recognition, Classification and Linking (MHERCL) benchmark[7] is a dataset for the historical EL task composed of manually annotated sentences selected from the English *Periodicals* module of the Polifonia Textual Corpus[8] (PTC), covering documents from

[7] https://github.com/polifonia-project/historical-entity-linking/tree/main/benchmark
[8] https://github.com/polifonia-project/Polifonia-Corpus

The Musical Times, 1873	
He also performed two of Mendelssohn (*Q46096*)'s	
Model	Top 10
Relik	**Felix Mendelssohn [Q46096]**, Francesco von Mendelssohn [Q1441287], Arnold Mendelssohn [Q537538], Fanny Mendelssohn [Q57286], Moses Mendelssohn [Q76997], ...
MPNet	Mendelssohn (disambiguation) [Q1794038], John Mendelsohn [Q1701059], List of compositions by Felix Mendelssohn [Q179039], Francesco von Mendelssohn [Q1441287], Robert Mendelsohn [Q7347623, ...
MPNet + ASP	Abraham Mendelssohn Bartholdy [Q70138], Arnold Mendelssohn [Q537538], Joseph Mendelssohn [Q96515], Fanny Mendelssohn [Q57286], Moses Mendelssohn [Q76997], ...
The Harmonicon, 1828	
Sontag (*Q64098*) left Francfort for Brussels on the Ist of December.	
Model	Top 10
Relik	Brussels [Q240], Sontag [Q47519541], Alan Sontag [Q945286], Susan Sontag [Q152824], Belfort [Q171545], ...
MPNet	Sontag [Q47519541], Sontag, MS [Q7562392], Sonbolabad [Q7560867], Sondor (disambiguation) [Q22349595], Frank Sontag [Q5489708], ...
MPNet + ASP	Sontag [Q47519541], Soner [Q962275], **Henriette Sontag [Q64098]**, Sonam [Q7560775], Ernst Sonntag [Q19661367], ...

Table 4
Examples of errors on MHERCL dataset. The mention to be linked is underlined in the sentence, while the correct candidate is highlighted in bold. Candidates are displayed in order by the bi-encoder similarity.

1823 to 1900. The issue date, provided in the metadata for each sentence of the dataset, is used as a reference point for the document date, while the type of each mention is annotated in the gold NERC data.

Results are shown in Table 3. Similar to the results of HIPE-2020, ReLiK performs best on the first 10 candidates but is consistently outperformed by the ASP-based method when more candidates are taken into account.

Table 4 analyses two example errors. In the first sentence, only ReLiK retrieves the correct candidate. It can be noted, however, that `List of compositions by Felix Mendelssohn` is retrieved, which is intuitively close to the correct mention. Similar to HIPE-2020's error, this is caused by the sentence embeddings and the linear projection. In the second sentence, ReLiK retrieves type implausible candidates, such as the city `Q240` (`Brussels`), and time implausible candidates, such as `Q945286` (`Alan Sontag`), born in 1946, and `Q152824` (`Susan Sontag`), born in 1933 (the issue date of the periodical from which the sentence is taken is 1828). On the other hand, the plausibility filters allow MPNet to retrieve the correct entity, which is `Q64098` (`Henriette Sontag`), mainly by leveraging the time constraint.

5.3. AjMC

The Sophocles' Ajax: a Commentary on Commentaries (AjMC)[9] is a dataset for historical EL composed of manually annotated documents from the 19th century containing commentaries of the Ajax Greek tragedy by Sophocles. Those documents contain a high density of named entities since they seek to give a complete summary of the tragedy while comparing it to other works.

Results are shown in Table 5. Similarly to previous datasets, ReLiK performs better on few documents considered and filtered bi-encoders perform best when additional entities are considered. However, it is worth noticing that some models reach a perfect recall within the first 50 candidates considered. Moreover, distill-RoBERTa reaches a perfect recall even without filters when 200 candidates are considered

[9] https://mromanello.github.io/ajax-multi-commentary/

Model		R@10	R@30	R@50	R@100	R@200	R@300
ReLiK [24]		**0.90**	0.93	0.93	0.94	0.99	1.00
MPNet [35]		0.38	0.50	0.52	0.94	0.98	1.00
	+ ASP	0.51	**0.96**	**1.00**	**1.00**	**1.00**	1.00
distill-RoBERTa [36]		0.29	0.47	0.51	0.92	**1.00**	1.00
	+ ASP	0.45	0.95	**1.00**	**1.00**	**1.00**	1.00
MiniLM [37]		0.23	0.39	0.39	0.50	0.98	1.00
	+ ASP	0.39	0.51	0.98	**1.00**	**1.00**	1.00

Table 5
Recall at different levels computed using different bi-encoders on AjMC dataset. Best results for each model are represented in bold.

<table>
<tr><td colspan="2">AjMC, 1881
but that the honour of Ajax (<i>Q172725</i>) and his race is in question.</td></tr>
<tr><th>Model</th><th>Top 10</th></tr>
<tr><td>Relik</td><td>AFC Ajax [Q81888], <b>Ajax the Great [Q172725]</b>, Ajax (horse) [Q4699589], Ajax II [Q4699606], Races [Q483225], ...</td></tr>
<tr><td>MPNet</td><td>Ajax [Q169527], Ajax Futebol Clube [Q4699604], Ajax Life [Q16835657], Ajax (cleaning product) [Q2828856], Jong Ajax [Q1770361], ...</td></tr>
<tr><td>MPNet + ASP</td><td>Ajax, Missouri [Q28103438], OpenAjax Alliance [Q1330650], AjaxView [Q4699582], ...</td></tr>
<tr><td colspan="2">AjMC, 1881
The article is not added to θεός elsewhere in Sophocles without special reason, and the conjecture of Schndw . (<i>Q70043</i>)</td></tr>
<tr><th>Model</th><th>Top 10</th></tr>
<tr><td>Relik</td><td>Acts of Andrew and Matthias [Q3374647], Antediluvian [Q4771131], Wḫdw [Q22936426], Phrygian Sibyl [Q928835], Gothic runes [Q1920146], ...</td></tr>
<tr><td>MPNet</td><td>SCW [Q353018], Schwebel [Q2254629], Scow (disambiguation) [Q59763596], Schmalkald [Q7431770], Schnepf [Q7431906], ...</td></tr>
<tr><td>MPNet + ASP</td><td>Wiener Schmäh [Q252069], Schnakenbach [Q2246950], Schwelge [Q832232], August Schmarsow [Q109788], <b>Friedrich Wilhelm Schneidewin [Q70043]</b>, ...</td></tr>
</table>

Table 6
Examples of errors on AjMC dataset. The mention to be linked is underlined in the sentence, while the correct candidate is highlighted in bold. Candidates are displayed in order by the bi-encoder similarity.

(differently than ReLiK). This might be attributed to the low number of documents in the dataset (167) and their nature. While the documents' are historical, the 19th-century language registry is similar enough to contemporary language that the language models can interpret it correctly.

In Table 6, we qualitatively report on two sample sentences. In the first sentence, ReLiK retrieves the correct candidate, while MPNet does not. However, it is possible to see how both models prefer popular entities (such as the football team *AFC Ajax*) despite their little relevance to the sentence. Additionally, it is possible to see that missing information influences the filters' performances. The type of `OpenAjax Alliance` (`ballot initiative`) is not mapped to any type within those of AjMC and is hence ignored, and there is no date assertion on Wikidata. The resulting entity has no type or year asserted in the program *P*; therefore is conservatively considered correct. In the second sentence, ReLiK struggles with OCR errors that impact the superficial mention of the named entity. Moreover, some of the retrieved examples are not plausible from the type perspective, such as `Q928835` (`Phrygian Sibyl`), an ancient Greek oracle, and `Q1920146` (`Gothic runic inscriptions`), the elder Futhark writings. On the other hand, the plausibility filters allow MPNet to exclude implausible entities and robustly retrieve the gold annotation, which is `Q70043` (`Friedrich_Wilhelm_Schneidewin`), a German classical scholar born in 1810.

5.4. TopRes19th

Model		R@10	R@30	R@50	R@100	R@200	R@300
ReLiK [24]		**0.83**	0.90	0.91	0.93	0.97	1.00
MPNet [35]		0.30	0.64	0.76	0.87	0.98	1.00
	+ ASP	0.73	**0.98**	0.99	**1.00**	**1.00**	1.00
distill-RoBERTa [36]		0.34	0.56	0.71	0.85	0.94	1.00
	+ ASP	0.59	0.72	0.99	**1.00**	**1.00**	1.00
MiniLM [37]		0.21	0.42	0.61	0.76	0.95	1.00
	+ ASP	0.62	0.95	**1.00**	**1.00**	**1.00**	1.00

Table 7
Recall at different levels computed using different bi-encoders on TopRes19th data. Best results for each model are represented in bold.

TopRes19th, 1867

THE MINING MARKET. London (*Q84*), Thursday Evening. There was very little business doing—dealers being busy with the fortnightly settlement.

Model	Top 10
Relik	**London [Q84]**, South Crofty [Q2304399], Wheal Metal [Q7991798], Wheal Eliza Mine [Q16903195], Wheal Vor [Q7991802], ...
MPNet	In London [Q6009848], London, Belgrade [Q6669759], London City [Q6670236], This Is London [Q7785842], Londons [Q261303], ...
MPNet + ASP	Little London, West Yorkshire [Q30006745], BBC London [Q902373], Education in London [Q5341069], .london [Q15928102], History of London [Q1126401], ...

TopRes19th, 1863

And that an AUDIT for the RESERVED and CHIEF RENTS for the Manor of Stayley, in the county of Chester (*Q23064*), will be holden at the Eagle Inn, in Stalybridge, on Thursday, the 7th day of May next, between the hours of Eleven and Two o clock, on which days the tenants are requested to pay their rents.

Model	Top 10
Relik	Chester [Q170263], Justice of Chester [Q616310], Earl of Chester [Q1277249], Earl of Warrington [Q5326386], Exchequer of Chester [Q5419617], ...
MPNet	Chester County [Q227112], Chester County Courthouse [Q1070703], Chester County History Center [Q19866503], New Chester [Q16462307], Diocese of Chester [Q543301], ...
MPNet + ASP	Chester Rural District [Q5093705], 1724 Chester Courthouse [Q4552563], Chester County, Pennsylvania [Q27840], Chester (town), Orange County, New York [Q2756901], **Cheshire [Q23064]**, ...

Table 8
Examples of errors on TopRes19th dataset. The mention to be linked is underlined in the sentence, while the correct candidate is highlighted in bold. Candidates are displayed in order by the bi-encoder similarity.

The TopRes19th[10] dataset is a collection of English historical newspaper articles from the British Library (18C-19C), whose annotations focus on toponyms entities aligned to their Wikidata entry.

Results are shown in Table 7. Coherently with the previous datasets, ReLiK performs best in the top 10 candidates, but it is outperformed by the sentence embedding models implementing ASP filters process when more candidates are considered. Notably, most of the bi-encoders quickly reach a perfect recall score, with MiniLM always retrieving the relevant candidate in the first 50 candidates. This is due to the specificity of the domain. Since only toponyms are taken into account, the program P always filters out every entity that does not represent a location. This optimal scenario showcases the high impact ASP constraints play when domain knowledge can be exploited.

[10] https://mromanello.github.io/ajax-multi-commentary/

Table 8 shows qualitative errors on two sentence samples. In the first sentence, similarly to the previous datasets, ReLiK performs better. The candidates retrieved by MPNet (with or without filters) are syntactically and conceptually similar to the target entity. In this case, the candidates seem to suffer from an *inverted* popularity bias: the popularity of the entity `London` should shadow other entities. Even though this is not desirable in general, it is beneficial for retrieving entities with little risk of ambiguity. In the second sentence, ReLiK struggles with type implausible candidates, such as the judicial position `Q616310` (`Justice of Chester`) and the noble title `Q1277249` (`Earl of Chester`). On the other hand, the plausibility filters allow MPNet to retrieve the gold annotation, which is `Q23064` (`Cheshire`), a ceremonial county in England, United Kingdom.

6. Discussion

The results of Section 5 clearly show that simple sentence embedding methods obtain competitive results when coupled with logical constraint and are able to consistently outperform more complex approaches when more than 10 candidates are considered.

We remark that the embedding models we tested are not trained to solve the entity linking task. Similarities between two vectors are optimized such that the whole content of the original documents is semantically similar. Nonetheless, by relying on a simple linear projection, the resulting similarity can retrieve all the relevant documents. Indeed, the sentence embedding methods we tested reached a perfect recall when 300 candidates were considered on all the datasets. This proves that, although they have not specifically been fine-tuned for entity-linking, they provide a solid base as bi-encoders. Using logical constraint is a robust method that can filter out most unrelated entities. While using an ASP solver introduces an overhead in the overall system, it provides a highly expressive language that enables domain experts to express and enforce trivial constraints that significantly improve the final results. Moreover, by relying on ASP it is possible to implement powerful automated reasoning techniques without significant effort that would be required for *ad-hoc* solutions. For instance, it is possible to enhance the type plausibility constraint by defining a taxonomy of types. An example application is to use ASP to automatically infer whether two types are related by a common ancestor and further filter the candidates based on it.

On the weak side, while the ASP solver provides great expressivity, it requires reliable facts. While structured KBs, such as Wikidata, can be used to that extent, they are not always straightforward to integrate with any dataset. In our experiments, we manually aligned Wikidata's types to the NERC classes of each dataset. Although this resulted in good performances, the alignment phase is not optimal. A more advanced method is required to align the structured KB to the structured information provided by the dataset.

Moreover, some tasks might not include metadata or structured information that can be aligned to Wikidata, such as open-ended question answering. An interesting approach is to automatically classify those datasets and exploit the probabilistic extensions of ASP to compute *probably* plausible entities - i.e. entities with a high joint probability of being relevant for the document and satisfying the constraints.

7. Conclusion and Future Work

This work presented a general method to integrate highly expressive logical constraints within a retrieval model. We experimented with this approach on several datasets annotated for entity linking and showed that the resulting method outperforms simple sentence embeddings and specialized methods.

Although our approach is general, it lends itself to tasks involving structured metadata or structured knowledge bases. Future works include extending this approach to documents that are only composed of unstructured content by employing automatic classification methods, such as automatically classifying named entities in a dataset that lacks gold NERC annotations. Additionally, given the promising results in the entity linking task, future works include extending SotA retrieval-based entity linkers, such as

ReLiK and EntQA, to leverage the logical constraints, both during inference and during the training phase, as an additional method to identify negative samples.

Moreover, testing on different tasks that are highly sensitive to recall in the retrieval phase, such as Question Answering, is an interesting extension, particularly on datasets whose question-answer are ranked according to the popularity of the entities contained in them [19, 20, 14].

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004746 and the FAIR – Future Artificial Intelligence Research foundation as part of the grant agreement MUR n. 341.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[2] S. Humeau, K. Shuster, M.-A. Lachaux, J. Weston, Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring, 2020. URL: https://arxiv.org/abs/1905.01969. arXiv:1905.01969.

[3] D. Gillick, S. Kulkarni, L. Lansing, A. Presta, J. Baldridge, E. Ie, D. Garcia-Olano, Learning dense representations for entity retrieval, in: M. Bansal, A. Villavicencio (Eds.), Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 528–537. URL: https://aclanthology.org/K19-1049. doi:10.18653/v1/K19-1049.

[4] N. Kandpal, H. Deng, A. Roberts, E. Wallace, C. Raffel, Large language models struggle to learn long-tail knowledge, in: Proceedings of the 40th International Conference on Machine Learning, ICML'23, JMLR.org, 2023.

[5] S. Shin, S.-W. Lee, H. Ahn, S. Kim, H. Kim, B. Kim, K. Cho, G. Lee, W. Park, J.-W. Ha, N. Sung, On the effect of pretraining corpora on in-context learning by a large-scale language model, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 5168–5186. URL: https://aclanthology.org/2022.naacl-main.380. doi:10.18653/v1/2022.naacl-main.380.

[6] Y. Razeghi, R. L. L. I. au2, M. Gardner, S. Singh, Impact of pretraining term frequencies on few-shot reasoning, 2022. URL: https://arxiv.org/abs/2202.07206. arXiv:2202.07206.

[7] X. Han, Y. Tsvetkov, Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data, 2022. URL: https://arxiv.org/abs/2205.12600. arXiv:2205.12600.

[8] V. Feldman, C. Zhang, What neural networks memorize and why: Discovering the long tail via influence estimation, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 2881–2891. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1e14bfe2714193e7af5abc64ecbd6b46-Paper.pdf.

[9] V. Feldman, Does learning require memorization? a short tale about a long tail, in: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 954–959. URL: https://doi.org/10.1145/3357713.3384290. doi:10.1145/3357713.3384290.

[10] N. Kandpal, E. Wallace, C. Raffel, Deduplicating training data mitigates privacy risks in language models, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceed-

ings of the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, PMLR, 2022, pp. 10697–10707. URL: https://proceedings.mlr.press/v162/kandpal22a.html.

[11] M. A. Stranisci, R. Damiano, E. Mensa, V. Patti, D. Radicioni, T. Caselli, WikiBio: a semantic resource for the intersectional analysis of biographical events, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 12370–12384. URL: https://aclanthology.org/2023.acl-long.691. doi:10.18653/v1/2023.acl-long.691.

[12] M. Stranisci, P.-L. Huguet Cabot, E. Bassignana, R. Navigli, Dissecting biases in relation extraction: A cross-dataset analysis on people's gender and origin, in: A. Faleńska, C. Basta, M. Costa-jussà, S. Goldfarb-Tarrant, D. Nozza (Eds.), Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 190–202. URL: https://aclanthology.org/2024.gebnlp-1.12.

[13] OpenAI, Chatgpt: Optimizing language models for dialogue, 2022. URL: https://archive.ph/4snnY.

[14] A. Graciotti, V. Presutti, R. Tripodi, Latent vs explicit knowledge representation: How ChatGPT answers questions about low-frequency entities, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 10172–10185. URL: https://aclanthology.org/2024.lrec-main.888.

[15] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.

[16] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum, Robust disambiguation of named entities in text, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 782–792. URL: https://aclanthology.org/D11-1072.

[17] A. Chen, P. Gudipati, S. Longpre, X. Ling, S. Singh, Evaluating entity disambiguation and the role of popularity in retrieval-based NLP, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4472–4485. doi:10.18653/v1/2021.acl-long.345.

[18] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Found. Trends Inf. Retr. 3 (2009) 333–389. URL: https://doi.org/10.1561/1500000019. doi:10.1561/1500000019.

[19] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi, When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories, 2023. arXiv:2212.10511.

[20] K. Sun, Y. E. Xu, H. Zha, Y. Liu, X. L. Dong, Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?, 2024. arXiv:2308.10168.

[21] L. Wu, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable Zero-shot Entity Linking with Dense Entity Retrieval, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6397–6407. doi:10.18653/v1/2020.emnlp-main.519.

[22] A. Graciotti, Knowledge extraction from multilingual and historical texts for advanced question answering, in: Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22st International Semantic Web Conference, ISWC 2023, Athens, Greece, 2023. URL: https://ceur-ws.org/Vol-3678/paper2.pdf.

[23] W. Zhang, W. Hua, K. Stratos, EntQA: Entity linking as question answering, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=US2rTP5nm_.

[24] R. Orlando, P.-L. Huguet Cabot, E. Barba, R. Navigli, ReLiK: Retrieve and LinK, fast and accurate entity linking and relation extraction on an academic budget, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics ACL 2024, Association for

Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024, pp. 14114–14132. URL: https://aclanthology.org/2024.findings-acl.839.

[25] S. Tedeschi, S. Conia, F. Cecconi, R. Navigli, Named Entity Recognition for Entity Linking: What works and what's next, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2584–2596. doi:10.18653/v1/2021.findings-emnlp.220.

[26] T. Ayoola, J. Fisher, A. Pierleoni, Improving entity disambiguation by reasoning over a knowledge base, in: M. Carpuat, M. de Marneffe, I. V. M. Ruíz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022, Association for Computational Linguistics, 2022, pp. 2899–2912. URL: https://doi.org/10.18653/v1/2022.naacl-main.210. doi:10.18653/V1/2022.NAACL-MAIN.210.

[27] M. Leszczynski, D. Y. Fu, M. F. Chen, C. Ré, Tabi: Type-aware bi-encoders for open-domain entity retrieval, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022, Association for Computational Linguistics, 2022, pp. 2147–2166. URL: https://doi.org/10.18653/v1/2022.findings-acl.169. doi:10.18653/V1/2022.FINDINGS-ACL.169.

[28] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (2011) 92–103. URL: https://doi.org/10.1145/2043174.2043195. doi:10.1145/2043174.2043195.

[29] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Clingo = ASP + control: Preliminary report, CoRR abs/1405.3694 (2014). URL: http://arxiv.org/abs/1405.3694. arXiv:1405.3694.

[30] C. Baral, M. Gelfond, J. N. Rushton, Probabilistic reasoning with answer sets, Theory Pract. Log. Program. 9 (2009) 57–144. URL: https://doi.org/10.1017/S1471068408003645. doi:10.1017/S1471068408003645.

[31] R. L. Geh, J. Gonçalves, I. C. Silveira, D. D. Mauá, F. G. Cozman, dpasp: A comprehensive differentiable probabilistic answer set programming environment for neurosymbolic learning and reasoning, CoRR abs/2308.02944 (2023). URL: https://doi.org/10.48550/arXiv.2308.02944. doi:10.48550/ARXIV.2308.02944. arXiv:2308.02944.

[32] R. Schwitter, Specifying and verbalising answer set programs in controlled natural language, Theory Pract. Log. Program. 18 (2018) 691–705. URL: https://doi.org/10.1017/S1471068418000327. doi:10.1017/S1471068418000327.

[33] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, S. Riedel, KILT: a benchmark for knowledge intensive language tasks, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 2523–2544. URL: https://aclanthology.org/2021.naacl-main.200. doi:10.18653/v1/2021.naacl-main.200.

[34] N. Reimers, I. Gurevych, 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, 2019. URL: https://arxiv.org/abs/1908.10084.

[35] K. Song, X. Tan, T. Qin, J. Lu, T. Liu, Mpnet: Masked and permuted pre-training for language understanding, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

[36] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).

[37] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Process-

ing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[38] M. Ehrmann, M. Romanello, A. Flückiger, S. Clematide, Overview of CLEF HIPE 2020: Named Entity Recognition and Linking on Historical Newspapers, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2020, p. 288–310. doi:10.1007/978-3-030-58219-7_21.

[39] M. Ehrmann, M. Romanello, A. Fluckiger, S. Clematide, Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers, in: Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, volume 2696, Thessaloniki, Greece, 2020, p. 38. doi:10.5281/zenodo.4117566.