September 2024

Enriching Large Language Models with Knowledge Graphs for Computational Literary Studies

Thora HAGEN [*,1], Janna OMELIYANENKO [*], Anton EHRMANNTRAUT, Andreas HOTHO, Albin ZEHE and Fotis JANNIDIS

Center for Artificial Intelligence and Data Science (CAIDAS), Julius-Maximilians-Universität Würzburg

ORCiD ID: Thora Hagen https://orcid.org/0000-0002-3731-6397, Janna Omeliyanenko https://orcid.org/0009-0006-2159-9413, Anton Ehrmanntraut https://orcid.org/0000-0001-6677-586X, Andreas Hotho https://orcid.org/0000-0002-0483-5772, Albin Zehe https://orcid.org/0000-0002-9472-0783, Fotis Jannidis https://orcid.org/0000-0001-6944-6113

Abstract. In this chapter, we explore the integration of knowledge graphs into large language models (LLMs) to enhance computational literary studies, a key subfield of computational humanities. Computational humanities have evolved significantly over the past decades, driven by the increasing digitization of cultural heritage data, advancements in computing power, and the development of advanced analytical methods. This digitization progress has particularly enabled the creation of knowledge graphs that capture the semantic relationships embedded in humanities data, including texts. These graphs represent structured knowledge that can enrich LLMs, enabling them to generate semantically rich representations even in domains with limited computational resources.

We explore how integrating knowledge graphs can enhance natural language processing (NLP) applications, specifically for the analysis of literary texts. To benefit the humanities and support the integration of KGs with LLMs for computational humanities, we discuss the specific content new KGs should ideally encompass. This necessitates a broader conceptualization of knowledge graphs, and supplies novel challenges for the field of knowledge graph creation, including for example diachronic concept alignment. Alongside this new perspective, we also propose the automatic creation of knowledge graphs from literary texts, such as graph-based plot representation, to allow for graph-based text analyses but also, again, create literary-informed LLMs through the integration of such graphs.

By demonstrating these techniques through the lens of computational literary studies, we illustrate the significant impact that knowledge graphs can have on enriching LLMs and advancing the humanities.

Keywords. knowledge graphs, natural language processing, computational humanities, computational literary studies

[*] Equal Contribution.

[1] Corresponding Author: Thora Hagen, `thora.hagen@uni-wuerzburg.de`.

1. Introduction

Since the availability of computers there has been research focusing on cultural heritage data: novels and other texts, paintings and other images, music and other audio data, feature films and other movies, excavation sites and other 3-dimensional objects. Though the digital humanities exist now for more than 60 years, only in the last 10–15 years enough material was digitized, enough computing power was available to process it, and methods sophisticated enough to analyze it had been developed. This confluence was the basis for a new sub-field: computational humanities (CH) [1]. CH are challenging because the humanities encompass numerous different disciplines that specialize in very different times and cultures. In each of these disciplines, cultural artifacts cannot be adequately studied without the relevant linguistic and cultural knowledge. A considerable amount of this knowledge has long been processed in dictionaries, lexicons, encyclopedias and other reference works. The comparatively small amount of surviving material further exacerbates the problem. In this situation, the possibility of significantly improving very modern research tools such as language models by means of knowledge graphs, into which dictionaries etc. have been transferred, represents an important opportunity to create semantically rich representations even for areas in which the prerequisites for modern computational access are not actually given.

In this chapter, we will use the example of computational literary studies, a sub-field of computational humanities, to show how these techniques are applied. In the first section we will clarify the notion of knowledge graphs and discuss what the term knowledge graphs means in the context of humanities data and the historical encyclopedia we study and use. In the second section, we will examine different ways of modeling texts as graphs, such as character networks, feature-based text similarity graphs, and plot structures. We argue that this kind of text modeling provides a novel perspective on literary text analysis, while also bearing the potential for language model integration. In the third section, we will review how knowledge graphs may be used in tandem with large language models. On the one hand, knowledge graphs can function as a resource to aid in language modeling. On the other hand, language models can also be employed for knowledge graph completion. Both techniques combined may for example aid in knowledge transfer from parameter-heavy LLMs to other models.

2. Knowledge Graphs in the Humanities

2.1. Definition

Definitions of what the term “knowledge graph” (KG) entails vary, not only across but within areas of research as well. Generally speaking, knowledge graphs are directed graphs with labeled nodes and edges. These are usually encoded as sets of triples, where head, relation, and tail of each triple correspond to human knowledge (often also referred to subject, predicate, and object). The nodes of the graph are also often referred to as entities, however the interpretation of what the term "entity" entails can be different. As per Hogan et al. [2], "nodes represent entities of interest and [...] edges represent relations between these entities." The authors also add that a knowledge graph is "intended to accumulate and convey knowledge of the real world," meaning entities in a knowledge

graphs are supposed to represent real world objects, and typically not concepts. As the term “knowledge graph” was first popularized by Google with the integration of multiple knowledge bases into their search engine ("things, not strings" [3]), it is predominantly understood that the knowledge contained in KGs corresponds to this kind of factual, real world knowledge (traditional definition, see e.g. [2,4,5,6]). However, another modernized definition of knowledge graphs has emerged over time, which strives to push the boundaries of KGs to also encompass any kind of human knowledge [7,8,9]. Knowledge graphs are accompanied by ontologies, which contain such formal constraints of the graph, for example which relations are allowed between which classes of entities. Crucial to any knowledge graph is also the disambiguation of subjects in the graph via unique identifiers, so that multiple subjects bearing the same name can be distinguished from each other. For machine readability, triples are typically represented in RDF format [10].

The definition provided by Hogan et al. [2] indicates that existing KGs and the methods associated with them in the computer science domain are still predominantly centered on real world objects. In the humanities, this practice is useful for museums, libraries, etc., where the main purpose of linked open data is to link cultural objects, however a broader approach is necessary to cover other aspects of the humanities as well. In this section, we explain why this modern approach to knowledge graphs is essential for creating new knowledge graphs tailored to the humanities. Across this chapter, we will highlight two specific approaches to knowledge graph creation which are based on our wider definition of the term: constructing a semantic network from encyclopedias to link concepts across time (Section 2.3), and generating graphs from text using a variety of inherent text characteristics (Section 3).

Broadly speaking, knowledge may be separated into the following two categories: factual knowledge (entity or world knowledge) and commonsense knowledge (e.g. agent-based knowledge or basic word meanings). Additionally, there is a domain-specific dimension to factual knowledge: some facts are general knowledge, such as *Berlin is the capital of Germany*, and other facts are native to a thematic domain, for example biomedical knowledge about drugs and diseases. Usually, language models capture human understanding through processing vast amounts of text. However, not all that humans know about the world is captured in this way, such as historical knowledge about people, concepts, customs, etc., that became less important or replaced over time. Knowledge graphs may be used to infuse additional knowledge into language models, as knowledge is expressed differently than through text (see Section 4, Adaptation). The humanities in particular have access to a multitude of semi-structured resources (*lexical-semantic resources* or *LSR*), such as dictionaries, encyclopedias, glossaries, which already convey knowledge in a similar fashion. These resources can further be refined into knowledge graphs.

Most experiments and benchmarks in NLP research concentrate on contemporary English. Here, research is often concerned with devising or refining methods, where input and evaluation data often play a secondary role. Because of that, KGs that are already available are often re-used across studies, because on the one hand it saves time and on the other hand, comparing methodological research becomes easier. Almost always, research concentrates on the KGs Wikidata (or subgraphs of it), ConceptNet, or WordNet (see Table 2 by Wei et al. [11] for an overview of data used). With that however, NLP sometimes overlooks historical varieties, other languages or textual specificities, all of

which lie at the core of humanities research. But leveraging resources of cultural heritage, which are often available to humanities researchers, can be an important step to fill these knowledge gaps for computationally processing and analyzing texts from resource-poor domains. These gaps are most mainly 1) time-specific or historical knowledge 2) language specific knowledge (endangered languages, dialects) and, for CLS in particular, 3) text domain knowledge. The humanities therefore care about all three knowledge types mentioned above, specific examples including:

- historical representation of facts and common sense, as they may change over time
- comprehensive vocabulary and word meaning representation for low-resource languages
- domain specific text knowledge about, for example, fictional worlds

This is why LSR are of special importance: They provide a thorough and oftentimes historical snapshot of knowledge about a language (dictionaries) or topics and facts (encyclopedias), and they can potentially function as KGs in the computational humanities. However, this means that humanities researchers need to create their own KGs from these sources to adopt the methods devised in NLP.

Because of the traditional KG definition, KG creation often relies on standard methods centered around named entities (NE recognition, NE linking, see [12,13,14]). KG creation for the humanities implies a slight shift towards concepts rather than entities, which is just one challenge that the field poses. In the following, we will outline the two main challenges of KG creation for the humanities [15]:

- diachronic data diversity (historicity and semantic change)
- synchronic data diversity (interdisciplinary knowledge).

2.2. Short Survey of Knowledge Graph Applications in the Humanities

2.2.1. From Structured Data to Knowledge Graphs

The topic of linked open data (LOD) in the humanities has actually been a long standing tradition beyond the usage of knowledge graphs. Especially in libraries, museums and archives, LOD catalogs have been a staple technology stack for the last decades, along with controlled vocabularies (such as VIAF[2]) and metadata sets (e.g. Dublin Core [16]). The overarching goal of LOD in the humanities is to promote accessibility and interoperability across diverse cultural domains and sources.[3] In humanities scholarship, traditionally, much research in the area has therefore been concerned with creating controlled vocabularies, classification schemes, or establishing cross-domain links [17,18].

Knowledge graphs and linked open data have now emerged as a new way to study and analyse cultural artifacts on a larger scale beyond simply digitally publishing data in LOD form [19]. Prominent application examples from the humanities are: character networks [20], correspondence networks and geospatial analysis [21,22], linked document collections through external or internal features [23,24,25], and metadata networks in general [26,27].

These examples demonstrate that the origin of knowledge graphs in the humanities can be very different: They may be built upon pre-structured knowledge as provided

[2] `https://viaf.org/`
[3] `https://lod-cloud.net/`

by libraries for example, or they may be built from the ground up from the basis of unstructured text. When building a knowledge graph from text, it depends on the use case of the resulting knowledge graph what kind of creation or knowledge extraction methods can be applied. Named entities and artifact metadata are also a staple of the humanities, where many pre-existing extraction methods and ontologies (most notably CIDOC CRM [28]) can be reused. But the use cases extend beyond that, such as building semantic networks like WordNet (which was manually created) or identifying specific inter-textual dependencies. Building a graph from text, be it to connect concepts across multiple texts (Section 2.3) or to better understand and analyze whole texts or their relationship with each other (Section 3) is where the most challenges arise for the field of knowledge graph creation in the humanities. We will discuss these challenges as well as opportunities in Section 2.3 on the basis of EncycNet.[4]

2.2.2. Interplay with the Computational Humanities

Structured data can take on different functions in the workflow of the computational humanities. On the one hand, as we already alluded to and which we will further discuss in Section 4, we may use knowledge graphs to enhance our computational model. On the other hand, we want to argue that structured data can also benefit the collection of other data and also allow model evaluation. Concepts in the humanities are less clear cut – for example the definition of a genre, the boundaries of a lexical field, or word meaning – in comparison to the tasks and interests that NLP prioritizes. As such, the subject of study may be harder to define in terms of representative data, which makes data collection or the establishment of a ground truth difficult. The same applies to model evaluation, where the (often manual and time consuming) creation of appropriate original datasets, is an indispensable step in every workflow.

This is where previous work on controlled vocabularies, taxonomies, schemas, etc., by humanists can also be exceptionally useful to the computational humanities. Thus, speaking from a broader point of view, we argue that all kinds of newly developed structured knowledge that pertain to the humanities can be profitable for symbolic representation in AI, and not just KGs. For instance, concerning the data collection, in a separate project we turned to a manually curated, digital dictionary of keywords in order to extract the lexical field of *privacy*, to be able to analyze how the whole field has semantically changed over time [29]. Concerning the model evaluation, we also developed new datasets for evaluating lexical semantics in LMs by harvesting word relations from other LSR and linked data sources – the German dictionary *Duden* and German Wiktionary – to aid in evaluating German pre-trained language models and word embeddings [30].

2.3. EncycNet: A Case Study for Knowledge Graph Creation

2.3.1. Challenges of EncycNet Knowledge Graph Creation

EncycNet aims to construct a historical German knowledge graph from nineteenth-century encyclopedias by computational means. The project offers one of the first openly accessible linked semantic data resources specifically for historical German. It is one of multiple projects within humanities research that uses LSR to create linked data. Among

[4] `https://encycnet.github.io/`

Table 1. Overview of the conversational encyclopedias in the EncycNet corpus.

Editor	Year	Description	# Entries	# Tokens
Brockhaus	1809–1811	Conversational encyclopedia	6 960	1 186 000
Herloßsohn	1834–1838	*Damen Conversations Lexikon*, specifically aimed at women	7 099	1 461 000
Brockhaus	1837–1841	Illustrated conversational encyclopedia	7 049	2 604 000
Herder	1854–1857	Conversational encyclopedia	39 755	2 256 000
Meyer	1905–1909	Comprehensive conversational encyclopedia	156 264	17 437 000
Brockhaus	1911	Conversational encyclopedia, paperback edition	82 780	2 434 000

other projects are, for example, LiLa[5] and PURA[6], which both aim to create diachronic linguistic knowledge graphs from historical thesauri for Latin and Greek, respectively. In comparison, EncycNet is focused on word meaning and semantics rather than word formation and linguistics, which is due to the nature of encyclopedias compared to thesauri or dictionaries. Common relations in semantic graphs include synonym and antonym word pairs for example, similar to some of the edges that can be found in ConceptNet or WordNet. An impression of the contents of EncycNet can be found in Figure 1, and an overview of the encyclopedias used can be seen in Table 1.

All these projects, which facilitate digital access to LSR, emphasize the importance of these cultural resources for the computational humanities. This also includes projects that make LSR digitally accessible in the first place, such as the *Nineteenth-Century Knowledge Project*[7] indexing the entries of the *Encyclopedia Britannica*, and the corpus curation for EncycNet [31]. These examples show that, even though LSR are analogue knowledge bases, they bear great potential for KG transformation, partly already due to their semi-structured nature.

The KG creation process can be separated into three steps: 1) ontology creation, that is, devising a conceptual schema that guides the design of the KG, 2) concept identification, and 3) relation extraction. In the particular case of EncycNet, broadly speaking, the encyclopedia entries function as subjects in the graph, while literals from the entries' texts and other entries function as objects. Concept identification in EncycNet thus also consists of alignment and disambiguation across encyclopedias. The challenges mentioned influence all steps of the KG creation process.

2.3.2. Diachronic Perspective: Historicity and Semantic Change

Most of the data within the humanities are time-sensitive, as simply most of human society and culture lie in the past. As already mentioned, NLP is mostly concerned with contemporary data, not only because KGs are more readily available, but evaluation data as well. In EncycNet, historicity plays a challenging part for KG creation, firstly, because the historical nature adds to data heterogeneity, and secondly because the semantic change of word meanings makes concept definition more difficult.

The genre of the encyclopedia has undergone a significant amount of change over the last few centuries. While striving to be a source of unbiased information today, ency-

[5] `https://lila-erc.eu`
[6] `https://pric.unive.it/projects/pura/home`
[7] `https://tu-plogan.github.io/`

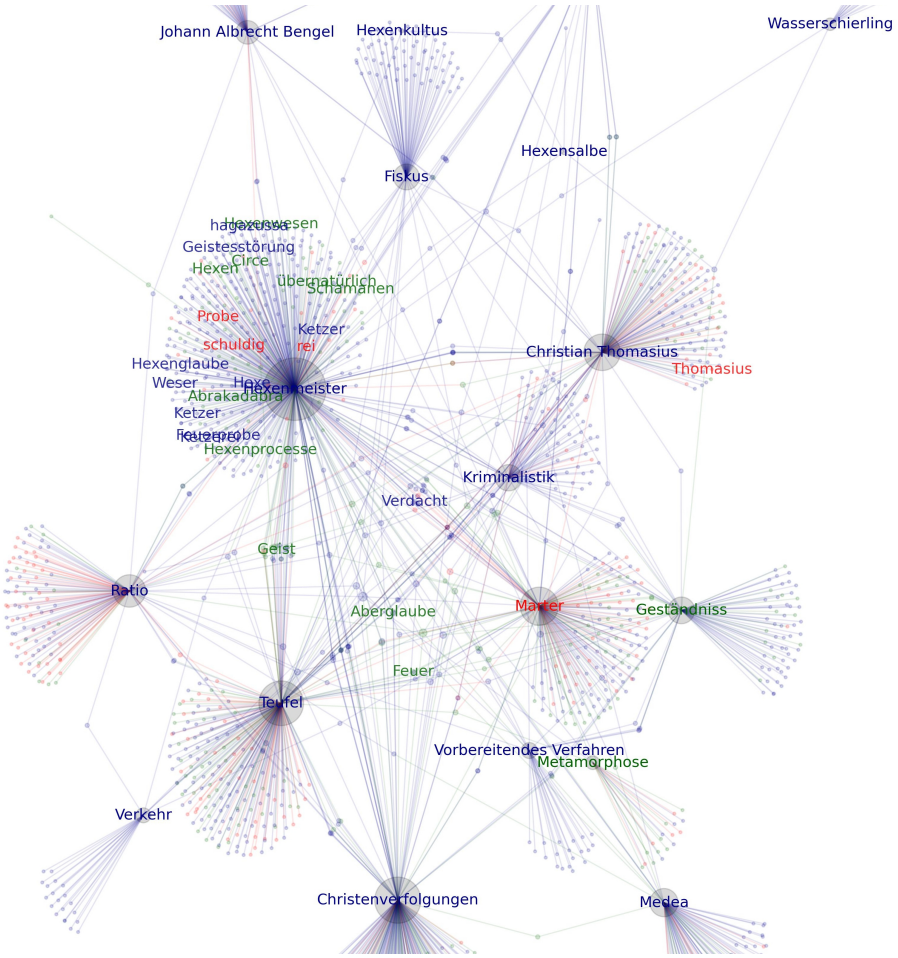


Figure 1. EncycNet subgraph representation of selected outgoing edges of central node *Hexe* (*witch*). Includes all neighbors of depth 1 and up to 30 more neighbors of depth 2 (for visibility purposes). Colors indicate different encyclopedias, i.e. where the edge or node was extracted from (green: Brockhaus 1837, red: Herder 1854, blue: Meyers 1905).

clopedias in the 19th century were highly diverse, even when discussing the same general concepts. This concerns the selection and organization of concepts, as well as the structure, scope, style and interpretation of individual concepts. Due to the fact that none of the encyclopedias can serve as an objective baseline, the alignment procedure would need to turn to another comprehensive and objective source. This is why in EncycNet, we chose to solve this step by aligning to Wikipedia/Wikidata as an intermediate step.

Entries of encyclopedias are made up of more than just named entities. This is why KG creation cannot turn to the method of entity linking in order to align concepts across sources. Rather, the procedure would need to adapt methods from word sense alignment (WSA) and word sense disambiguation (WSD) [32,33,34]. Here, three types can be distinguished: text-to-text, text-to-graph, and graph-to-graph. Therefore, alignment procedures are often either based on text or graph similarity measures. We modeled the align-

ment as a text-to-text similarity problem: concepts sharing the most similar definitions, given either by the entries gloss or the summary part of Wikipedia, would be aligned. We first selected candidates for alignment through Wikipedia's search function, targeting Wikipedia's disambiguation pages in particular. For our model, we performed continued training on TSDAE [35], a LM trained specifically to produce sentence or paragraph embeddings, using Wikipedia and encyclopedia entries to adapt to the text genre. We then generated gloss embeddings for each concept and its corresponding candidates with TSDAE. Using cosine distance, we aligned concepts with its most similar candidate. We manually annotated a balanced evaluation dataset of ambiguous entries in the encyclopedias with their ground truth Wikipedia pages, which then, for our automated retrieval, resulted in a quite high median number of candidates (10.5). We achieved an overall F1 of 0.66 on our testset, where we found that disambiguation is generally harder for named entities (0.47 for people, 0.58 for locations) than for abstract concepts (0.66) and objects (0.70) [36].

The concept alignment across multiple encyclopedias – spanning now two centuries when including Wikipedia – bears significant challenges. First, as concepts have undergone semantic change, it becomes difficult to assess whether a concept should actually be aligned with its counterpart from a different time period. Definitions could have become broader or narrower, now perhaps even merging two previously different concepts (sometimes resulting in archaic words) or splitting into two entirely. In the *witch* example from Figure 1 for instance, both *Hexenmeister* and *Hexe* were aligned to the same Wikidata entry, one being the male and one being the female version of a witch. It remains to be discussed how to asses concept boundaries and their shifts computationally, and where the semantic threshold for cross-temporal alignment lies.

Historical variation is additionally noticeable in terms of orthography. For instance, the contemporary word form *Freiheit* appears as historical orthographic variant *Freyheit*, similarly *Tier* vs. *Thier*, *Konzert* vs. *Concert*, etc. In EncycNet, we performed orthographic normalization using the rule-based tool CAB, developed and maintained by the DTA (*Deutsches Textarchiv*). In fact, orthographic text normalization remains an open problem. An evaluation on parallel data between historical literary texts (c. 1750–1900) and their normalized editions shows that even sequence-to-sequence Transformer-based normalizers still make about one error per one hundred words [37].[8] In fact, the impact of orthographic varieties on downstream NLP/CLS tasks is largely unknown, and further investigations in this area are required.

2.3.3. Synchronic Perspective: Cross-Domain Knowledge

The encyclopedias in the EncycNet corpus are so-called "conversational" encyclopedias (*Konversationslexika*), which means that they encompass any kind of knowledge. Consequently, facts, common sense and certain domain-specific aspects must be taken into account simultaneously when creating the KG, at least when following a holistic approach. To make such knowledge extraction feasible, generic classes of entries need to be identified as a first step. The classes of EncycNet consist of two types of entities (people, locations) and three types of concepts (objects, abstract concepts, fictional concepts), which were retrieved after the Wikidata alignment from Wikidata's taxonomy.

[8] `https://github.com/aehrm/hybrid_textnorm`

In a second step, the most important relations for each class may be selected for knowledge extraction. This ensures not giving preferential treatment to some (at times highly specific) knowledge domains, simply because they seem easy to extract. Even if knowledge graphs are not manually curated, the choice of computational methods or knowledge to extract can lead to biased data, especially when handling nearly all human knowledge at the same time. To give some examples, this applies to chemical notations, currency exchanges, weight conversions, or highly nested hierarchies, which can be found for languages for example.

The interdisciplinary approach for knowledge graph creation also influences the choice of the underlying ontology. There exist multiple ontologies already which aim to support linked data from cultural heritage sources. The ontology CIDOC CRM [28] is the most prominent one and is often referenced in the digital humanities. The data model is extremely vast, however the focus lies on the traditional KG definition, meaning on entities and events. To add lexemes to KG ontologies (in contrast to focusing on named entities only) has just recently been a point of interest in the community, such as the lexical import of WordNet into Wikidata. Other ontologies such as OntoLex aim to support the exact representation of LSR to KG (e.g. one entry references another), however this data model does not take semantic relationships between lexemes into account. Overall, these standards are well suited for sub-areas of encyclopedias, but the complete representation of all content that EncycNet strives for, including mainly lexical properties, is not entirely possible. Besides the advantages for alignment, this is also why we chose the most extensive data model, Wikidata, where we only included properties relevant to EncycNet.

2.3.4. Outlook and Discussion

EncycNet is one of the first openly available historical German knowledge graphs,[9] striving to be a new data source for DH and NLP alike. In future work, we will further analyze how the data can impact LM integration, in particular when it comes to the detection of semantic change for historical German LMs.

To summarize, it can be said that the field of knowledge graphs is already making a stronger impact in favour of the humanities, but there is still room for improvement in many areas. For one, there is a greater need in terms of a machine-readable data representation for time-sensitive data. Even though RDF* and many other projects exist to adapt RDF (reification in particular) to allow for time annotations, many of the ideas are still preliminary or are not consistently supported formats throughout libraries or applications. Second, the de-facto standard for cultural heritage modeling, CIDOC CRM, is extremely complex and vast. Currently, this is why it is necessary to build domain specific extensions of the ontology [38], meaning it is not suited for a cross-domain knowledge approach. Wikidata on the other hand provides an easy-to-use and extensive data model, where projects can rather select properties and concepts of interest instead of extending the model, however it lacks standardization. Compromising between standardization and generalization when developing ontologies in cultural heritage therefore remains to be a challenge in the field. Third, development of methods surrounding concepts and disambiguation of word senses will have to move into the foreground, not just for the sake of

[9] `https://encycnet.github.io/`; RDF knowledge graph available at `http://dx.doi.org/10.5281/zenodo.10219192`

KG creation. Many WSD techniques rely on linked data resources as "sense inventories" [39]. To create a historical linked data source with sense disambiguated concepts at this point in time means relying on contemporary data, and in turn integrating potentially false assumptions. Especially for resources like encyclopedias, where every sense of a word is listed and in turn is a highly ambiguous resource, we need to refine methods that do not rely on external data. This entails for example, as stated in Section 2.3.2, a clear cut model of when a word sense changes meaning.

3. Text as Graphs

In the previous section, we have discussed the challenges and chances in creation of KGs from existing semantic resources for CLS and provided a specific example, with the knowledge graph EncycNet. In addition to this semi-manual creation from existing resources, it is also feasible – and in many cases very promising – to extract graph representations from the texts themselves. On the one hand, this enables the use of graph-based methods to further process the extracted information. On the other hand, it also gives us the chance to directly feed the information we have extracted so far back into our models using, among others, the methods we will describe in Section 4 – leading to a kind of feedback loop, where the model can iteratively gain a better understanding of the texts by reading through them multiple times and re-using the knowledge the model has gained in previous passes. One example where graph representations are directly suitable for literature are character networks. These networks can serve as a kind of condensed representation of a text. A very simple character network could show which characters interact in the text, while more sophisticated versions could include information about friendship or family relations. This is also a prime example for the feedback loop: A model can extract information about the characters in a text to a character graph and then directly incorporate this graph to gain a better understanding of the text.

In the remainder of this section, we will describe existing experiments towards the use of graph representations for the processing of literary texts. We start by introducing research on representing characters and static character networks using graph neural networks, before introducing an approach to provide a temporal partitioning of the texts in scenes and finally working towards combining these two approaches to represent texts as a sequence of scenes represented by character networks, among other features. Our intended next step in future work is to finalize this combination by applying temporal graph neural networks to the sequence of character networks, building a more comprehensive representation of the plot of a novel.

3.1. Character Networks as a Representation for Literary Characters

In previous work [40], we have extracted character networks from different texts in Tolkien's Legendarium in a semi-automated way, creating a joint character network over all texts. We have shown that applying graph neural networks to this representation leads to better representations of the characters than just using word embeddings based on the texts.

We use the English full text of The Lord of the Rings (consisting of three volumes: The Fellowship of the Ring, The Two Towers and The Return of the King), The Hobbit

and The Silmarillion. The texts are pre-processed using BookNLP[10] and co-references are resolved in a semi-automatic way, only resolving explicit named mentions (e.g., "Peregrin Took", "Sam Gamgee") due to the difficulty of full co-reference resolution on literary texts, especially due to its long document length [41]. A joint character network for all texts is then constructed from co-occurrences of characters in the same sentence.

We evaluate the suitability of Graph Neural Networks for the representation of the character network in two downstream tasks: character classification and link prediction. In the first task, we aim to build a classifier that is able to detect which book (Hobbit, Silmarillion or Lord of the Rings) a character is most strongly affiliated with. This shows the model's ability to represent the structure of the character networks. The same approach can also be used to classify characters with regard to their role in a text, for example main character, mentor or antagonist, when a larger collection of texts is available with these roles annotated. In the second task, we want to build a model that is able to infer whether a link (i.e., an interaction) between two characters is missing from the network. This can be used to reconstruct co-occurrences in the texts that have been missed due to errors in the co-reference resolution.

For both of these tasks, we compare character embeddings created using different types of graph neural networks with Word2Vec embeddings of the character names. These word embeddings are trained on Tolkien's novels, but lack the structural information from the character networks. We show that Graph Convolutional Networks using Word2Vec embeddings as additional node features outperform pure Word2Vec embeddings in both tasks, raising the F1-score for node classification from 79.45 % to 92.32 % and the ROC/AUC score for link prediction from 78.16 % to 81.67 %. This proves that the structural information encoded in the character networks is indeed useful for representing the characters in the texts.

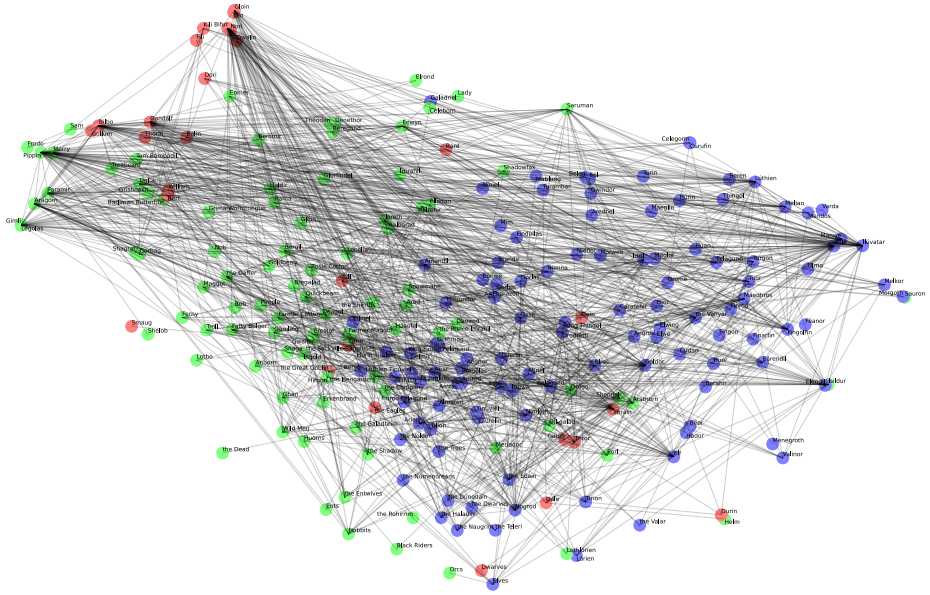
In addition to these quantitative evaluations, we also use both Word2Vec embeddings and embeddings produced from graph neural networks to build visualizations of the character networks. Figure 2 shows the resulting visualizations. We can see that the GNN-based visualization provides a clear separation between the three books, while the Word2Vec-based visualization mixes characters from all books.

Overall, our experiments in this paper show that NLP methods and graph based methods can be combined very effectively: We can use NLP methods to extract graphs from texts and then use these graphs to build better representations of the entities in the text.

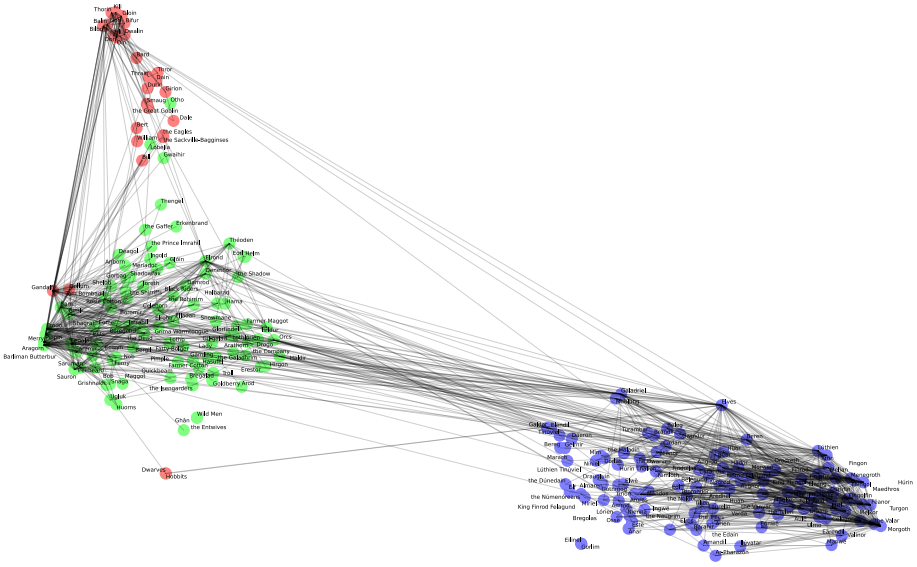
3.2. Segmentation of Novels into Scenes

While the static representations used above [40] can provide a good overview of the story's structure and characters, they ignore one of the most important aspects of literature: the development of the story over time. In order to model this development, a segmentation of the text into coherent parts is necessary. This may be done on a purely structural level, for example by using paragraphs or pages as a unit for segmentation. However, we argue that a segmentation strategy grounded in literary theory is desirable. Therefore, we have proposed the task of scene segmentation, where a text is to be split into segments that are coherent across four dimensions: time, space, action and character constellation [43]. In short, this means that a scene should not contain major jumps in

[10] `https://github.com/booknlp/booknlp`.



(a) Latent space embedding of characters obtained by applying the word embedding Word2Vec to the whole text corpus



(b) Latent space embeddings obtained by the Graph Convolutional Network using Word2Vec character embeddings as additional node features

Figure 2. Comparison of two latent space embeddings of characters in Tolkien's Legendarium using (a) Word2Vec and (b) the hidden layer activations of a Graph Convolutional Network, where nodes carry additional features that correspond to Word2Vec embeddings. Node colors indicate the work in which characters appear most frequently (red: The Hobbit, green: The Lord of the Rings, blue: The Silmarillion). Edges represent character co-occurences in the text. Two-dimensional visualizations were generated using the dimensionality reduction technique t-SNE [42].

time, should take place in a single location, contain a fixed set of characters and narrate one main course of action.

After initially discussing the task [44], we have formally introduced it along with an annotated dataset [43] and organized a shared task to collect new approaches [45]. The most successful approach so far was one of the contributions to the shared task, introduced by Kurfalı and Wirén [46]. It models the problem as a sequential sentence classification task, where each sentence is classified as either a scene border or no scene border. The model takes as input a piece of text consisting of multiple sentences, which are interleaved by `SEP` tokens between the sentences, and the resulting representation is passed through a pre-trained encoder model, for example BERT. The resulting contextualized representations of the `SEP` tokens are then used as input to a multi-layer perceptron, which classifies each `SEP` token independently as either a scene border or no scene border.

We were able to slightly improve the performance of this model by using a BERT-model with additional domain-adaptation on literary texts, as described in Sec. 4.1 [47]. While this approach constitutes the current state of the art in scene segmentation, the performance still leaves room for improvements: The model is evaluated on two different test sets, one constructed from the same type of texts as the training data (i.e., dime novels) and one from high-brow literature texts that are expected to be more challenging to understand and therefore segment. The model reaches F1-scores of 0.40 and 0.35 on the in- and out of-domain data, respectively.

3.3. Plot as Temporal Graphs

In [48], we made a step towards combining these two approaches to generate a computational representation of the abstract concept of the *plot* of a novel. We applied a scene segmentation algorithm to a set of dime novels from the genres horror and romance to partition each of them as a sequence of scenes. We then extracted, for each scene, a set of features to form an explicit representation of the scene and compile these scene representations to a time series representing a novel. This representation is comprised of several factors:

- the characters that are present in the scene;
- triples of (subject, verb, object) representing the action in the scene;
- valence and arousal values extracted using a sentiment lexicon;
- association with a few simple types of interactions: fighting, erotic actions and talking;
- character-level values for valence and arousal; and
- a distribution over topics in the scene extracted using topic modeling.

We evaluate the suitability of this time series representation to capture the plot of a novel via three proxy tasks: First, we compare the similarity of novels from different genres. We find that our representation is able to capture the difference between horror and romance novels well, assigning a higher distance between novels of different genres on average. However, we show that this simple task can also be solved by static representations of the texts, as for example tf-idf-representations.

To further validate our approach, we construct synthetic evaluation datasets by permuting the order of scenes in the novels. If our representation is capable of capturing

the usual structure of the plot, the similarity between the unmodified novels should be higher than between the shuffled novels. Our experiments show that this is indeed the case, indicating that the model takes into account the order of scenes.

3.4. Open Challenges

Throughout our experiments towards the conversion of novels into structured representations, we have identified several research challenges that need to be addressed in order to improve the suitability of these representations.

Co-Reference Resolution The most obvious use case for graph representations in CLS is the creation of character networks. However, these can only be accurately extracted when character references can be detected and resolved reliably. High-quality co-reference resolution is therefore an extremely important prerequisite for these character network-based representations of novels. So far, we have relied either on the performance of existing co-reference resolution algorithms being sufficient [48] or fallen back to semi-automatic resolution [40]. However, this is one of the points where our previously mentioned feedback loop is promising: We can collect knowledge about characters in a graph representation and then use this information to improve the performance of co-reference resolution. For example, since the model has access to the information which characters are friends or enemies, it can therefore better reason about the likelihood of interactions between the characters. We are currently actively working towards the development of models implementing this feedback loop.

Detecting Present Characters Another challenge for the extraction of character networks is the detection of which characters are present in the current scene. While a good enough co-reference resolution can tell us which characters are mentioned, it is still necessary to determine whether a character is actually present in the scene or not (e.g., is only mentioned in conversation or even just used as a reference, or if the character is not mentioned at all). This can be partially addressed by considering only characters that are referenced outside of direct speeches, but a more comprehensive detection is still an open research question.

Temporal Graph Models While the use of graph neural networks (GNNs) has seen great development in the last couple of years, most of these GNNs are still focused on representing static graphs. While implementations for temporal graph neural networks are also available [49], these have so far not shown good results in our experiments. We are on collaboration with researchers from the field of graph learning to develop approaches that are better suited for the representation of literary texts.

Temporal Order of Scenes The order of scenes in a text in many cases does not follow the chronological order within the narration – flashbacks or parallel narrations are very common tools. For a more accurate representation of novels as a sequence of scenes, we are currently working towards automatically detecting the chronological ordering of scenes.

4. Improving NLP Models with Knowledge Graph Data

Pre-trained language models (LMs) trained on vast amounts of textual data – broadly understood here to include both contextualized models such as BERT, and also decontextu-

alized ones such as word embeddings – have demonstrated powerful abilities in modeling language semantics and achieving state-of-the-art performance on most natural language processing downstream tasks. The effectiveness of a pre-trained LM for the CLS task depends heavily on its ability to capture not only contemporary language semantics and knowledge but also historical and time-specific contexts. Tasks such as the automated creation of social networks of characters from a specific book depend on the performance of named entity recognition (NER) and named entity linking (NEL) techniques, which, in turn, rely on the quality of the model's language representation of the book's time period. Due to the limitation of low-resource data, domain-specific language models trained on domain-specific text corpora often struggle to accurately capture the semantics of the domain-specific language. Consequently, they perform significantly worse on downstream tasks than models trained on large, general language corpora.

Literary studies in general have only a small amount of training text data available, making it challenging to train LM or static word embeddings with a good language representation of a given domain or time period. Therefore, there is a need for solutions that enable learning a good semantic language representation under the constraints of a small training dataset.

Recent research has shown the effectiveness of adaptation of pre-trained language models, initially trained on large general language corpora, to domain-specific language by further pre-training them on smaller, domain-specific text corpora [50,51,52]. Complementing the work on domain adaptation through pre-training on domain-specific text corpora, a number of research works have shown the potential of integrating knowledge graphs that explicitly model semantic relations, into pre-trained language model to improve its domain semantics and performance on domain-specific tasks [53,54,55,56,57].

In this chapter, we will present our work on investigating the potential of integrating knowledge graphs into language models to improve their semantics and performance on downstream tasks within the field of CLS. First, we will review the applicability of domain adaption to the CLS domain by continuing the pre-training the model on small CLS relevant historical domain corpora to improve performance on downstream tasks of named entity recognition and the detection of speech rendition. Secondly, we will demonstrate the potential of applying knowledge graph information to adjust non-contextualized word embeddings trained in low-resource settings using retrofitting and fusion methods to improve their semantic representation. Next, we will show how pre-trained language models, trained on vast amounts of textual data, can enrich the knowledge graph with information on the semantic relation strength between concepts. We will use this weighted knowledge graph to adjust non-contextualized embeddings through retrofitting, further enhancing their semantic representations. Finally, we will propose an approach for continually integrating newly occurring knowledge graphs into pre-trained language models, while avoiding catastrophic forgetting.

To demonstrate these approaches, we apply them in this chapter to the well-established BERT language model [58], as this model has been extensively used in knowledge graph research [59,60]. Nevertheless, we note that the methods themselves are model-agnostic and discuss strategies to apply them to alternative language models in Sec. 5.

4.1. Language Model Adaptation to Historical Domains

Inspired by [51], we investigate the applicability of further pre-training on (unlabeled) domain-specific datasets to the low-resource CLS domains of interest, prior to fine-tuning the language model to CLS-specific tasks. This adjustment aims to improve the quality of the model language representation and the performance of CLS tasks in domain-specific contexts.

4.1.1. Methodology

In our work, we utilized a German BERT model [58] pre-trained on German texts from heterogeneous domains and a DistilBERT [61] model based on German BERT. To adapt the generally pre-trained models to the desired domain or temporal language, we employed a two-step procedure. In the first step, we followed the methodology from [51], where the model is continuously trained further on domain-specific text data, task-specific (unlabeled) text data, or a combination of both. In the second step, the models pre-trained in step one are fine-tuned on two downstream tasks: named entity recognition (NER) and detection of speech rendition. Additionally, we alternatively pre-trained an ELECTRA [62] model from scratch on domain-specific textual data and downstream task data. We chose this model because it requires less training data and time while providing a performance similar to that of BERT. The pre-training parameters for the models, as well as further details on the experimental setup, can be found in [63].

4.1.2. Named Entity Recognition

Named entity recognition is an information extraction downstream task that aims to extract and classify the named entities mentioned in unstructured textual data. In our experiments, we have used the German Corpus of Novels (DROC) [64], which brings together 90 annotated fragments of German language novels.

4.1.3. Detection of Speech Rendition

Detection of speech rendition is a classification task that aims to identify and classify types of speech within unstructured textual data. For our experiments, we used the labeled dataset "Redewiedergabe" [65], which categorizes speech into four possible classes: direct, indirect, free indirect, and reported. Detailed information on the dataset and the task can be found in [63].

4.1.4. Results

We compared our models to those without further fine-tuning on domain-specific datasets. The results of our experiments are presented in Table 2. As expected, due to low-resource constraints, the ELECTRA baseline trained from scratch showed low performance. The results indicate that continued pre-training on domain-specific textual data and downstream task textual data consistently improves performance over models directly trained on downstream tasks. While using textual data and downstream task data separately enhances performance, the combination of both further boosts the performance of the models. We hypothesize [63] that precisely this pre-training on the (unlabeled) task test data may allow the language model to build a better representation of the test data, helping in accurately solving the downstream task. This is particularly use-

ful in the context of CLS, where the datasets we are interested in are often very small and closed. In summary, the results demonstrate the potential of applying continued pre-training for domain adaptation to improve language representations for domains of interest in CLS.

Table 2. Results of 10-fold cross-validation on the DROC NER and speech rendition tasks per model. Reported numbers are F1 scores. *Base* refers to models fine-tuned without additional pre-training steps. *Domain* models are pre-trained on the domain corpus and *Task* models on the downstream task text corpus. *Task+Domain* refers to the combination of both. [63]

	DROC NER			Speech Rendition		
Model	**BERT**	**DistilBERT**	**ELECTRA**	**BERT**	**DistilBERT**	**ELECTRA**
Base	.859 (.032)	.838 (.32)	.791 (0.03)	.538 (.086)	.455 (.04)	.373 (.057)
Domain	.863 (.041)	.854 (.028)	n.a.	.637 (.234)	.557 (.356)	n.a.
Task	.882 (.039)	.853 (.031)	n.a.	.587 (.403)	.568 (.18)	n.a.
Task+Domain	**.927 (.029)**	.893 (.043)	n.a.	**.691 (.136)**	.655 (.132)	n.a.

4.2. KGs for Word Embeddings for Lexical Semantics

In the previous section, we demonstrated that continued pre-training on small domain-specific textual data has the potential to improve language representation in CLS domains. Complementary to continued fine-tuning on textual data, many research works have shown that pre-training on structured knowledge such as KGs can be used to further improve the performance of static and contextual language representation [66,57,53,56,54,55]. Some semantic word relations occur infrequently in text data and are even more rare in domains with limited textual data sources such as CLS. This makes learning a good semantic word representation even more challenging. KGs on the other hand, as described in Sec. 2, models these semantic relations explicitly. By integrating this information into static or contextualized word representations, the semantic relations that are difficult to capture directly from the text data can be represented.

As discussed in 2.1, CLS studies focus on cultural heritage, which encompasses structured information sources such as encyclopedias and dictionaries. These sources contain historical and time-specific knowledge and information about word semantics and semantic relations between these words. These structured information sources can be used to create historical KGs such as EncycNet (Sec. 2.3). This makes it attractive to apply these solutions to these high quality CLS data to learn better semantic word representation in low resource scenario of CLS.

In our work, we propose to integrate structured semantic information from dictionaries into static word embeddings by using established applications of 1) retrofitting and 2) fusion to improve the semantic representation of words when data resources are limited.

4.2.1. Methodology

In this section, we describe in detail the approaches we propose to apply.

Retrofitting Retrofitting [67] proposes to utilize the relational information KGs to refine existing word embeddings by forcing the words that are connected to each other in KG to have a similar representation. The main idea is to re-adjust the embedding vector of each word so that it is both close to its original embedding vector and close to the embeddings of all connected words in the KG.

Formally, the retrofitting refinement objective can be formulated to minimize

$$\Psi(Q) = \sum_{i \in V} \left[\alpha_i ||q_i - \hat{q}_i||^2 + \sum_{(i,j) \in E} \beta_{i,j} ||q_i - q_j||^2 \right] \quad (1)$$

where q_i and $\hat{q}_i$ denote the adjusted and original embeddings vectors of word or entity i from a set of entities V of a KG respectively, and $\beta_{i,j}$ represents the weight of the edge (i, j) connecting word i with word j in the KG. The hyper-parameter α_i determines how close q_i should stay to its original vector $\hat{q}_i$. In our work, an optimized version [68] of the retrofitting algorithm was used.

Fusion The fusion approach [69] proposes an alternative solution for adjusting word representation embeddings using semantics represented in KGs. This solution trains two types of embeddings. KG embeddings for given KGs using established approaches such as TransE [70], RotatE [71] on the one hand, and word embeddings on text data such as Word2Vec [72], FastText [73] on the other hand. The embedding of the word is concatenated with the respective embedding of the entity on the KG and then reduced to the desired dimensionality using methods such as PCA, AVG, SVD [69]. In our work, we used RotatE to train KG embeddings and FastText for word embeddings. PCA was used for dimensionality reduction.

4.2.2. Datasets

To train the FastText word embedding, we used a sample of German OSCAR [74] dataset. German is not a low-resource language, but to simulate a low resource setting, we randomly selected a small subset of sentences from OSCAR, resulting in a dataset of about 20MB (about 3.6M tokens, 47,000 types). Since we evaluated general linguistic performance in terms of word meaning representation, rather than performance for a specific thematic domain, we proceeded with this type of corpus creation to represent a low-resource language. We used GermaNet [75] as a KG source to train RotatE embeddings. GermaNet describes semantic relations between words and is therefore suitable for our study. In [76], we have described detailed information on data pre-processing and the resulting data statistics.

4.2.3. Evaluation Setup

We evaluated the resulting KG adjusted word embeddings on the established tasks of word similarity and word relatedness by using multiple German datasets: Schm280 [77] (translated WordSim-353 [78]), SimLex999 [79] (translated from [80]), ZG222 [81] as well as Gur65 (translated RG-65 [82]) and Gur350 [83]. The datasets contain word pairs with annotated human ratings indicating how similar or related these words are. In our experiments, we used the cosine similarity of the resulting word embeddings to calculate the similarity/relatedness score, and computed correlation to gold human scores. We compared the results of KG-adapted word embeddings with FastText embeddings without KG integration.

Table 3. Performance of the adapted word embeddings on the selected data sets (Spearman correlations between the cosine similarities of the word vector pairs and the human ratings). For the FastText model, the standard deviations of 15 runs are also given. Retro+Fusion averages the cosine similarities of both models.

	SimLex-999	**Schm280**	**ZG222**	**Gur65**	**Gur350**
# instances	825	242	120	49	237
FastText	0.224 (0.004)	0.495 (0.01)	0.299 (0.01)	0.320 (0.03)	0.653 (0.01)
Retro	0.267	0.512	0.291	0.490	0.607
Retro (only synsets)	0.253	0.487	0.273	0.374	0.652
Fusion	0.250	0.484	**0.347**	0.426	**0.666**
Retro+Fusion	**0.278**	**0.537**	0.337	**0.497**	0.660

4.2.4. Results

In Table 3, it can be seen that the KG-adapted embeddings consistently perform better than the word embeddings without KG integration, emphasizing the applicability and potential of using KG for CLS studies with low resource constraints.

4.3. LM4KG

The information represented in KGs can be used to adapt language representations of word embeddings and LMs to new domains or to understand new knowledge [66,54,53,55]. As shown in the previous section, this information is very useful when learning word representations under the constraint of limited training data. An important aspect of this type of adaptation is understanding the reasonability and strength of relations between concepts in these KGs. As specified in Eq. (1) the retrofitting approach uses weight information of edges $\beta_{i,j}$ that connects two entities i and j in a KG, when adjusting respective word embedding representation. Therefore, the quality of the adjusted word embeddings when using these approaches strongly depends on the quality of the information about the relation weights. Not all relations in KGs are equally strong, some are more meaningful and relevant than others. For example, in ConceptNet [66], a common sense KG that contains general world knowledge, considering the entity "word" would return "God" and "language" as related entities. In most contexts, however, the term “language” is more appropriate therefore this information should be represented accordingly as a weight in the KG [84]. In addition to factual and common sense KGs, character graphs [40] also describe relations between characters in a book that vary greatly in strength and meaningfulness. However, most of available KGs lack explicit information about the strength or weights of relations, so that each relation is considered equally important. Therefore, an automatic solution for assessing the strength of relations in KGs is of high interest.

It has been shown that pre-trained LMs contain a certain amount of factual knowledge [85] and are even able to perform question answering to a certain extent without being explicitly trained on this task [86].

In this section, we present our *REWEIGHT* [84] approach, which leverages this inherent knowledge available in a pre-trained LMs to assess the strength of the relations in KGs. This LM enriched KG is then utilized to generate more accurate word embeddings, demonstrating the potential of both utilizing LM to improve KGs and applying these enhanced KGs to better model language semantics. The *REWEIGHT* methodology consists

of several steps. For each relation (edge) between two words in a given KG, we construct a natural language sentence representation using a manually defined set of templates and automated grammar correction. These generated sentences are then fed into a pre-trained LM, and the output perplexity of the model is estimated. To map the estimated perplexity to the weight range of the original graph, we apply a transformation where a high edge weight represents a strong semantic relation between two words. Finally, the estimated weights are inserted into the original KG, resulting in an enriched knowledge resource that reflects the strength of its relations. By applying retrofitting to this enriched KG, we obtained static word embeddings with improved semantic representations.

4.3.1. Methodology

In this section, we describe in detail the methodology of our proposed *REWEIGHT* solution for the automatic weighting of relations in KGs, and application of the enhanced KG to produce word embeddings with improved semantic representation. In addition, we describe the datasets used as well as the evaluation setup.

Sentence Generation We aim to evaluate the weight of relations in a knowledge graph (KG) using a pre-trained language model (LM). To do this, we first need to represent the KG's relations as natural language sentences. We have manually defined a set of templates to map the relations between graph entities to sentences. For example, a "DefinedAs" relation between words A and B in the ConceptNet KG is converted to the sentence *"A is defined as B."* The complete list of templates is provided in [84]. This simple template-based solution can produce grammatically poor sentences for some relations. Since LMs have been shown to encode both syntactic and semantic information, and we want to use the LM to assess only the semantics, it is important to mitigate the influence of poor grammar. To achieve this, we applied a grammar correction model to all generated sentences, resulting in equivalent sentences with improved syntax.

Weight Generation After constructing a sentence representation for each relation in the KG, we use a LM to assess the strength of these relations. To achieve this, we evaluate the meaningfulness of the generated sentences using perplexity. We assume that a pre-trained LM will assign high perplexity to sentences describing questionable or uncommon relations. In our work, we used the bidirectional language model BERT [58]. It has been shown that the common definition of perplexity is not applicable to bidirectional LMs [87]. Therefore, in our experiments, we used their perplexity approximation to calculate the score for each sentence.

Weight Transformation To convert the estimated weights to the weight range of the original graph, where a high weight represents a strong relation between two words, we propose two approaches: $\textit{REWEIGHT}_{\text{light}}$ and $\textit{REWEIGHT}_{\text{mod}}$. Here, we only provide a description of $\textit{REWEIGHT}_{\text{light}}$ and refer to [84] for the alternative approach.

$\textit{REWEIGHT}_{\textit{light}}$ This variant of the perplexities transformation, calculates the weight in the range of the original KG for a relation r as follows

$$\beta_{\text{RWL}}(r) := \frac{\gamma_{\max}}{pp(r)} \tag{2}$$

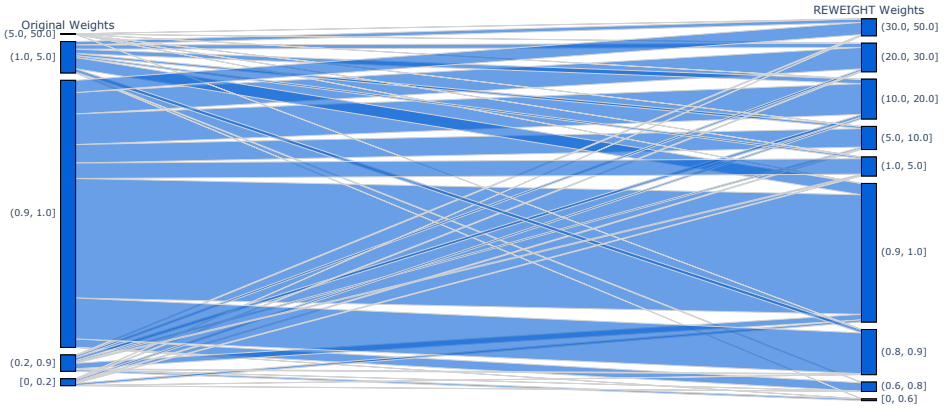


Figure 3. Sankey diagram showing the change of ConceptNet weights before and after REWEIGHTing. The blocks on the left resp. right side represent sets of ConceptNet relations, clustered by their respective weights, before and after REWEIGHTing, into interval. The streams between the blocks represent the changes in the compositions of the clusters.

where γ_{max} denotes the maximum weight in the original KG and $pp(r)$ the perplexity assigned to a sentence describing relation r by an LM. The resulting weights are then used to replace the edge weights of the KG.

4.3.2. Experiments

We applied our methodology to the ConceptNet KG. Although KG relations have weights, these weights reflect the trustworthiness of the source from which the relation was extracted rather than the semantic strength of the relationship between two words. The changes in weight distribution before and after applying our method are illustrated in Figure 3.

To evaluate the quality of the new LM-generated weights, we used the ConceptNet KG in a retrofitting process similar to Sec. 4.2 to adjust word embeddings semantics, following the experiments proposed by Speer et al. [66].

We take the word embeddings retrofitted from the original ConceptNet weights (*NumberBatch*) as baseline, and then compare them to retrofitted word embeddings which have been obtained from our LM-based reweighting approach $REWEIGHT_{\text{light}}$ and $REWEIGHT_{\text{mod}}$. The quality of the adjusted embeddings was evaluated on word similarity and relatedness tasks using multiple established datasets: MEN3000 [88], Rare Words [89], MTurk-771 [90], WS353 [78], SimLex999 [80], and Semeval17-2a [91]. Unlike the experiment from previous Section 4.2, these datasets are in English.

4.3.3. Results

The results of our experiments can be found in Table 4. The results demonstrate that embeddings adjusted using ConceptNet with weights from our method perform consistently better than the baseline. This highlights the potential synergy between KGs and LMs in refining semantic representations, which is crucial for the CLS tasks.

Table 4. Spearman correlation of embeddings generated through retrofitting with different KGs on multiple word similarity datasets. Significant difference to baseline through Fischer's z-transformation with $^{\dagger}p < 0.01$, $^{\S}p < 0.05$.

Group	Embedding	MEN3000	RW	MTurk	WS353	SemEval	SimLex	Average
Baseline	*Numberbatch*	0.872	0.630	0.822	0.833	0.779	**0.633**	0.762
Ours	$REWEIGHT_{\text{light}}$	0.877	†**0.663**	0.827	0.840	**0.783**	**0.633**	**0.770**
	$REWEIGHT_{\text{mod}}$	†**0.881**	§0.651	**0.828**	**0.845**	0.780	0.618	0.767

4.4. CapsKG

The knowledge landscape is constantly evolving, with new information emerging regularly in the form of new facts, KGs, genres, books, and character graphs. CLS studies on cultural heritage focus on structured knowledge sources, such as encyclopedias and dictionaries that are specific to various domains or time periods. These studies consistently provide high-quality information sources, such as historical KG EncycNet (Section 2.3). Consequently, language models need to be able to seamlessly integrate and update to accommodate this new knowledge input to better represent the semantics of a particular domain, time, genre, or even book character networks.

However, previous solutions for iterative fine-tuning often face the major obstacle of catastrophic forgetting [92]. This phenomenon refers to the tendency of models to forget previously learned information when they are updated with new data. Although there are solutions that aim to prevent catastrophic forgetting, they often have the disadvantage that they cannot effectively transfer the knowledge previously learned by the model, which is crucial in a low-resource domain such as CLS. As CLS studies continuously create new knowledge sources like EncycNet, it is important to have solutions that integrate new knowledge into language models without causing catastrophic forgetting. Such solutions should also enable access to previously learned information, which is especially beneficial for low-resource domains. In our CapsKG study, we propose the use of capsule architecture to address these problems. CapsKG enables the continual integration of new knowledge into LMs while simultaneously overcoming the problem of catastrophic forgetting.

4.4.1. Methodology

In this section, we describe the details of our proposed solution CapsKG [93] for the continual integration of newly occurring knowledge into a language model. The main goal of CapsKG is the continuous integration of knowledge into an LM when new knowledge sources become available. As such, CapsKG faces two main challenges. On the one hand, the model needs to make knowledge from previously integrated sources available when integrating new information, to prevent learning of redundancies and enable knowledge transfer. On the other hand, previously integrated knowledge also needs to be protected from catastrophic forgetting during the continual integration of new information sources. To address these challenges, CapsKG uses two main components. A source-specific feature extractor module extracts relevant information for the currently integrated knowledge source and leverages a routing mechanism to make the relevant information from previously integrated sources available. Additionally, a shared adapter module provides

joint access of model parameters to all sources while preventing catastrophic forgetting of previously learned sources through a gating mechanism.

To introduce CapsKG in detail, we will first discuss the input representation and the training objective for integrating knowledge sources into the LM. Next, we will explore both the source-specific feature extractor module with routing mechanism that enables the reuse of information and the shared adapter module.

Input Representation and Training Strategy To integrate new knowledge into a language model l, we employed a well-established approach. This method entails training the model on the link prediction task using the prompting method [94,59], which involves converting knowledge triples t of a KG into natural language sentences, referred to as prompts. Although automated methods for this conversion exist, e.g. [95], we chose to use the method from Sec. 4.3.1, which relies on manually created sentence templates for simplicity. For example, given the triple *"Alexandria _location_location_containedby Egypt"*, the generated sentence is *"Alexandria is located in Egypt"*. To train the model on this knowledge, we applied a masking training strategy similar to Bosselut et al. [94] and Fichtel et al. [59]. We masked the object or subject in each sentence, resulting in the prompt p *"Alexandria is located in [MASK]"* and trained the model to predict the missing entity *"[MASK]"*. The model was trained using cross-entropy loss. Given a conversion function *prompt* that transforms an incomplete knowledge graph triple (e,r) with kept entity e into a prompt p with a masked entity e', and a pre-trained LM l, the link prediction using LM can be formally defined as follows:

$$prom(e,r) \rightarrow p, \qquad l(p) \rightarrow e' \tag{3}$$

where r is the relation type and e' is predicted by the LM.

Source-Specific Feature Extractor with Routing Mechanism The task of the source-specific feature extractor with routing mechanism is to extract features relevant to each knowledge source and enable knowledge transfer between similar sources to complement each other. To achieve this, the module contains knowledge source-specific extractor capsules, which are a set of small fully connected layers that extract relevant low-level features. A transfer capsule layer represents the transferable features extracted by the extractor capsules. A special similarity based routing mechanism, consisting of several steps, is used to transfer features between lower-layer extractor capsules and higher-layer transfer capsules. This mechanism enables knowledge exchange between similar sources while avoiding exchanges between dissimilar sources, thus preventing the introduction of noise that could negatively influence representations. The detailed mathematical definition of this module can be found in [93].

Shared Adapter Module To address the problem of catastrophic forgetting when continually learning new knowledge, the capsule architecture proposes utilizing an adapter solution [96] within the shared adapter module. Adapters consist of small trainable neural layers inserted into the layers of a pre-trained language models l. They are trained to learn relevant features for a new task, such as link prediction [97] for a particular KG with ID k, while the base language model l remains frozen. This solution allows the language model l to learn new knowledge while preserving its original language representations. Formally, the adapter can be defined as follows:

$$h'^{(k)} = f_c(f_d(h^{(k)})) + h^{(k)}. \tag{4}$$

where $h^{(k)} \in \mathbb{R}^{d_s \times d_e}$ denotes the output of an intermediate layer of the language model l, where d_s represents the sequence length and d_e represents the dimension of the base model's hidden state output. This output serves as the input to the adapter. f_c and f_d define the fully connected layers of the adapter. Adapter layers map the input to an intermediate adapter dimensionality, which can be freely chosen, and then back to the hidden dimensionality space of the model. Finally, the intermediate output of the original base model is added via skip connections. To facilitate further knowledge transfer, the capsule architecture utilizes a single adapter for all knowledge sources. To prevent catastrophic forgetting in the adapter when training with a new knowledge source, a masking strategy is applied. For each KG' ID k and each adapter layer, a mask $m^{(k)}$ is calculated from the knowledge graph embedding $e^{(k)}$ based on the KG's ID. The masks have the same dimensionality as the respective adapter layer. The KG embeddings $e^{(k)}$ are learned during training. The mask is calculated by converting the knowledge graph embedding into a pseudo-gating function, applying the Sigmoid activation function σ and a scale hyper-parameter s. Formally, this is defined as follows:

$$m^{(k)} = \sigma(se^{(k)}) \tag{5}$$

The scale hyper-parameter s is a positive scalar that is gradually increased during the training procedure, forcing the learned mask to contain values closer to 0 or 1. The obtained pseudo-binary masks $m_d^{(k)}$ and $m_c^{(k)}$ for the respective adapter layers f_c and f_d are multiplied element-wise with these adapter layers, calculating the output of the shared adapter module as follows:

$$h'^{(k)} = f_c(f_d(h^{(k)}) \otimes m_d^{(k)}) \otimes m_c^{(k)} + h^{(k)} \tag{6}$$

Applying masks of previously learned knowledge sources to the gradients during training on new one, preserves the information learned from previous KGs, thus preventing catastrophic forgetting of already learned knowledge. Additionally, other KGs can reuse already reserved neurons in the forward path, further facilitating knowledge sharing between different KGs.

4.4.2. Experiments and Datasets

Since CLS studies focus on cultural heritage including structured information sources and analyses of the social network of characters, the task of link prediction is highly relevant in CLS domain. We trained and evaluated our solution on the link prediction task using three established English KG datasets: YAGO [98,99], WN18 [70], and FreeBase [100]. The preprocessing steps and resulting dataset statistics for the three English KG datasets can be found in [93]. To also show the potential of the proposed solution for CLS, we conducted experiments with a German historical knowledge graph, EncycNet [36].

For the German historical EncycNet dataset, we split the graph into 19 sub-KGs, each containing triples of a specific relation type. Each subgraph represents a distinct knowledge source that is newly introduced and needs to be integrated into the LM. This

Table 5. Dataset characteristics of the used EncycNet dataset after preprocessing according to [93].

dataset	relation types	entities	train triples	dev triples	test triples	total
EncycNet	19	671341	275680	59263	58843	393786

allowed us to integrate these KGs into the LM separately and demonstrate the benefits of reusing previously learned knowledge when learning new knowledge. The resulting subgraphs were then randomly split into training, validation, and test datasets with a ratio of 70 %, 15 %, and 15 %, respectively. Following this, all preprocessing steps from [93] were applied to the resulting subgraph splits. Statistics for the EncycNet dataset after preprocessing are provided in Table 5.

We evaluated the following models in our experiments:

$\text{BERT}_{\text{frozen}}$ A BERT base model that is prompted without any fine tuning.

BERT An independent BERT base model is trained and evaluated for each relation type [59].

BERT-CL A single BERT base model is iteratively trained for all relation types and evaluated once at the end.

Adapter An independent BERT base model with frozen parameters and one trainable Adapter is trained and evaluated for each relation type [101].

Adapter-CL A frozen BERT base model with a single trainable Adapter is iteratively trained for all relation types and evaluated once at the end.

CapsKG The CapsKG model [93] is iteratively trained for all relation types and evaluated once at the end.

4.4.3. Results

The experimental results are presented in Table 6. Next to the general common sense knowledge graphs of WN18, YAGO, and FB15k, our proposed solution for continual knowledge integration also consistently outperforms all baselines on the EncycNet dataset. This demonstrates its potential for effectively integrating new knowledge into LM in the CLS domain.

For a closer interpretation of CapsKG's performance on the EncycNet dataset, we report the model's performance on the individual relation types contained in EncycNet in Table 7. Since our goal is to integrate newly acquired knowledge into the LM, we

Table 6. Results for link prediction on WN18, YAGO, FB15k with 5 random seeds, and EncycNet with 10 random seeds, reporting mean and standard deviation of Hits@1 performance.

Model/Dataset	**WN18**	**YAGO**	**FB15k**	**EncycNet**
$\text{BERT}_{\text{frozen}}$	10.7 (0.0)	27.0 (0.0)	5.5 (0.0)	7.4 (0.0)
BERT	25.8 (0.5)	48.0 (0.1)	34.7 (0.7)	38.93 (0.5)
Adapter	26.4 (0.5)	48.7 (0.4)	35.6 (0.2)	39.93 (0.3)
BERT-CL	20.2 (1.2)	40.4 (4.0)	11.4 (3.2)	23.6 (5.9)
Adapter-CL	18.4 (1.2)	41.5 (0.7)	16.3 (1.4)	25.0 (5.6)
CapsKG	**27.4 (0.4)**	**49.4 (0.3)**	**36.1 (0.3)**	**40.7 (0.3)**

Table 7. Results for link prediction on EncycNet on individual relation types for a frozen BERT model and the full CapsKG method, reporting mean and standard deviation of Hits@1 performance for CapsKG over 10 random seeds.

Relation Name	**BERT$_{\text{frozen}}$**		**CapsKG**	
	train	test	train	test
Komposition	1.90	0.60	86.89 (7.30)	26.40 (0.49)
Nomenklatorischer Typ	0.00	0.00	95.97 (1.33)	93.89 (0.76)
Relevante Person	3.60	3.60	28.27 (1.16)	24.98 (0.25)
Alternativer Name	2.30	3.20	49.86 (4.45)	39.82 (2.27)
Etymon	6.60	6.20	62.62 (5.94)	38.61 (2.78)
Beschrieben-in	0.00	0.00	51.01 (10.66)	45.73 (1.86)
Wörtliche Übersetzung	3.90	4.20	32.27 (5.61)	16.16 (1.14)
Synonym	4.40	4.60	59.73 (8.15)	49.67 (1.98)
Affiliation	0.40	0.40	31.54 (3.43)	27.63 (0.86)
Ist-eine	4.20	4.00	54.79 (5.27)	48.66 (0.41)
Taxon Hypernym	76.10	76.90	94.35 (1.06)	93.26 (0.52)
Klasse	0.00	0.00	75.56 (0.70)	75.07 (0.45)
Übersetzung	0.50	0.30	19.15 (6.38)	11.80 (0.48)
Definition	4.50	4.20	41.20 (2.82)	39.30 (0.82)
Sprache	21.30	22.70	77.77 (0.46)	75.31 (0.43)
Aspekt-von	0.50	1.30	27.43 (5.30)	13.93 (1.44)
Bedeutender Autor	0.60	0.70	9.69 (0.16)	10.36 (0.23)
Verwandter Begriff	0.50	0.60	20.58 (2.26)	17.36 (0.27)
Bedeutender Ort	7.50	7.90	29.51 (1.15)	25.58 (0.41)

also report the model's performance on the training split to assess how well it accommodates new information. The results show that some relation types can be learned better by the model than others. Specifically, general concepts related to common sense abstractions such as hypernyms (Taxon Hypernym), the language of a concept (Sprache), or the biological nomenclatural type (Nomenklatorischer Typ) appear to be well represented within the model. Conversely, highly specific knowledge related to specific persons (Relevante Person) or public literary figures (Bedeutender Autor) appears to result in lower accuracies. Some relation types, such as compounds (Komposition), are challenging for the LM to generalize but can be effectively accommodated by the model. Still, the proposed approach is able to learn a large variety of historical knowledge, as evidenced by the comparison to the standard BERT$_{\text{frozen}}$ model. This regular pre-trained BERT model that was not fine-tuned on the knowledge graph performs exceptionally poorly on most relation types, indicating that integrating available knowledge from knowledge graphs provides significant benefits to language models.

5. Outlook: Applications of Neurosymbolic Systems and Knowledge Graphs

Enhancing LLMs with structured knowledge is only one step towards creating more stable knowledge representations within LLMs. On the one hand, LLMs such as Llama or GPT possess quite an extensive and thorough repository of knowledge. On the other hand, significant drawbacks of LLMs are their unpredictability when prompted as well

as a lack of transparency, not only when it comes to the data that is actually contained in them, but also potentially additional knowledge filtering steps the user might not know about. In comparison, KGs offer a straight-forward knowledge representation, however they only cover a fraction of how humans actually store and process knowledge. To integrate KGs and LLMs means seeking to remedy the drawbacks of one another to some extent: transparent (but shallow) knowledge from KGs with vast (but unpredictable) knowledge from LLMs.

In this chapter, we demonstrated multiple strategies for integrating information from knowledge graphs into language models. We showed how existing word embeddings may be updated according to a knowledge graph through retrofitting and fusion. Additionally, we demonstrated CapsKG, an approach for continuously integrating knowledge graphs into language models through parameter-efficient fine-tuning. Lastly, we also showcased the LM4KG method for extracting information contained in a language model and integrating it into an existing knowledge graph. While our demonstrations of these approaches rely on established encoder-based language models, we note that these approaches can also be adapted to currently popular large decoder-based language models. As the intuition behind the LM4KG method is directly reliant on a generative intuition of model perplexity, decoder-based language models are directly applicable and may even be better suitable for assessing viable candidate concepts as they enable a direct perplexity calculation due to their unidirectional nature. Additionally, current LLMs can be prompted to focus on specific historical periods or domains, facilitating a more detailed assessment of triple strength based on the given context. As these models can process longer input texts, prompts can be extended with relevant documents, such as books, enabling the model to evaluate triple strength based on information specific to those documents. For CapsKG, the adapter-based methodology used for fine-tuning the model to the knowledge graph closely corresponds to current parameter-efficient fine-tuning strategies of large decoder-based language models and is directly transferable to these architectures.

Overall, in this chapter, we argued that LLMs can benefit from knowledge graphs representing historical and cultural heritage as well as literary text characteristics, thus advancing the field of computational literary studies. Conversely, knowledge graph creation and completion can also benefit from the latest advancements in LLMs.

However, the research area of KG and LLM integration can only advance state-of-the-art language modeling up to a certain degree, as the grounding problem stretches far wider than what KGs can actually provide. As Mollo and Millière argue, the key to to creating a LM that is better connected to the real world may not necessarily lie in the kind of data provided, but rather in the nature of the model's learning process [102]. Future work in the field will need to focus on integrating novel data and data structures with appropriate and innovative learning procedures.

Acknowledgements

This publication was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the project LitBERT, project no. 529659926.[11]

[11] `https://gepris.dfg.de/gepris/projekt/529659926`.

References

[1] Biemann C, Crane GR, Fellbaum CD, Mehler A. Computational Humanities. Bridging the gap between Computer Science and Digital Humanities (Dagstuhl Seminar 14301). Dagstuhl Reports. 2014;4(7):80-111. doi:10.4230/DAGREP.4.7.80.

[2] Hogan A, Blomqvist E, Cochez M, D'amato C, Melo GD, Gutierrez C, et al. Knowledge Graphs. ACM Computing Survey. 2021;54(4):71:1-71:37. doi:10.1145/3447772.

[3] Singhal A. Introducing the knowledge graph: Things, not strings; 2012. Google Blog. Available from: `https://blog.google/products/search/introducing-knowledge-graph-things-not`.

[4] Nickel M, Murphy K, Tresp V, Gabrilovich E. A review of relational machine learning for knowledge graphs. Proceedings of the IEEE. 2015;104(1):11-33. doi:10.1109/JPROC.2015.2483592.

[5] Cimiano P, Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web. 2017;8(3):489-508. doi:10.3233/SW-160218.

[6] Abu-Salih B. Domain-specific knowledge graphs: A survey. Journal of Network and Computer Applications. 2021;185. Article 103076. doi:10.1016/j.jnca.2021.103076.

[7] Peng C, Xia F, Naseriparsa M, Osborne F. Knowledge graphs: Opportunities and challenges. Artificial Intelligence Review. 2023;56(11):13071-102. doi:10.1007/s10462-023-10465-9.

[8] Ji S, Pan S, Cambria E, Marttinen P, Philip SY. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE transactions on neural networks and learning systems. 2021;33(2):494-514. doi:10.1109/TNNLS.2021.3070843.

[9] Zhang H, Liu X, Pan H, Song Y, Leung CWK. ASER: A Large-scale Eventuality Knowledge Graph. In: Proceedings of The Web Conference 2020. Association for Computing Machinery; 2020. p. 201–211. doi:10.1145/3366423.3380107.

[10] Cyganiak R, Wood D, Lanthaler M. RDF 1.1 Concepts and Abstract Syntax; 2014. World Wide Web Consortium (W3C) Recommendation. Available from: `https://www.w3.org/TR/rdf11-concepts/`.

[11] Wei X, Wang S, Zhang D, Bhatia P, Arnold A. Knowledge enhanced pretrained language models: A compreshensive survey; 2021. ArXiv preprint. doi:10.48550/arXiv.2110.08455.

[12] Kejriwal M. Domain-specific Knowledge Graph Construction. Springer; 2019. doi:10.1007/978-3-030-12375-8.

[13] Zhong L, Wu J, Li Q, Peng H, Wu X. A comprehensive survey on automatic knowledge graph construction. ACM Computing Surveys. 2023;56(4):1-62. doi:10.1145/3618295.

[14] Zeng K, Li C, Hou L, Li J, Feng L. A comprehensive survey of entity alignment for knowledge graphs. AI Open. 2021;2:1-13. doi:10.1016/j.aiopen.2021.02.002.

[15] Hagen T. Von A bis Z: Überlegungen zur Erstellung eines Wissensgraphen aus historischen Enzyklopädien. In: Busch A, Trilcke P, editors. Abstracts zur 9. Jahrestagung des Verbands Digital Humanities im deutschsprachigen Raum e.V. (DHd2023); 2023. doi:10.5281/zenodo.7688632.

[16] Apps A. Guidelines for encoding bibliographic citation information in dublin core metadata; 2005. Dublin Core Metadata Initiative.

[17] Haslhofer B, Isaac A, Simon R. Knowledge graphs in the libraries and digital humanities domain; 2018. ArXiv preprint. doi:10.48550/arXiv.1803.03198.

[18] Hawkins A. Archives, Linked Data and the Digital Humanities: Increasing Access to Digitised and Born-digital Archives via the Semantic Web. Archival Science. 2022;22(3):319-44. doi:10.1007/s10502-021-09381-0.

[19] Hyvönen E. Using the Semantic Web in digital humanities: Shift from data publishing to data-analysis and serendipitous knowledge discovery. Semantic Web. 2020;11(1):187-93. doi:10.3233/SW-190386.

[20] Labatut V, Bost X. Extraction and analysis of fictional character networks: A survey. ACM Computing Surveys. 2019;52(5):1-40. doi:10.1145/3344548.

[21] Ryan YC, Ahnert SE. The measure of the archive: The robustness of network analysis in early modern correspondence. Journal of Cultural Analytics. 2021;6(3):57-88. doi:10.22148/001c.25943.

[22] Windhager F, Salisu S, Liem J, Mayr E. In: Dammann F, Kremer D, editors. The Knowledge Graph as a Data Sculpture: Visualising Arts and Humanities Data with Maps, Graphs, and Sets over Time. Bielefeld University Press; 2024. p. 113-34. doi:10.1515/9783839469187-007.

[23] Schöch C, Hinzmann M, Röttgermann J, Dietz K, Klee A. Smart Modelling for Literary History. International Journal of Humanities and Arts Computing. 2022;16(1):78-93. doi:10.3366/ijhac.2022.0278.

[24] Zamani M, Tejedor A, Vogl M, Kräutli F, Valleriani M, Kantz H. Evolution and transformation of early modern cosmological knowledge: A network study. Scientific Reports. 2020;10(1):article 19822.

doi:doi.org/10.1038/s41598-020-76916-3.

[25] Barber M. Reading and understanding text reuse data through graphs: a case study on pre-modern Arabic book history and historiography. In: Andrews T, Diehr F, Efer T, Kuczera A, van Zundert J, editors. Proceedings of the 6th International Conference on Graphs and Networks in the Humanities; 2022. Available from: https://graphentechnologien.hypotheses.org/files/2022/01/Reading_and_understanding_text_reuse_data_through_graphs_etc-Barber.pdf.

[26] Jain N, Múnera AS, Lomaeva M, Streit J, Thormeyer S, Schmidt P, et al. Generating Domain-Specific Knowledge Graphs: Challenges with Open Information Extraction. In: Tiwari S, Mihindukulasooriya N, Osborne F, Kontokostas D, D'Souza J, Kejriwal M, editors. Proceedings of the 1st International Workshop on Knowledge Graph Generation From Text and the 1st International Workshop on Modular Knowledge (TEXT2KG/MK @ ESWC). vol. 3184 of CEUR Workshop Proceedings; 2022. p. 52-69. Available from: https://ceur-ws.org/Vol-3184/TEXT2KG_Paper_4.pdf.

[27] Verhoeven D, Burrows T. Building the Australian Knowledge Graph: HuNI (Humanities Networked Infrastructure). In: Andrews T, Diehr F, Efer T, Kuczera A, van Zundert J, editors. Proceedings of the 6th International Conference on Graphs and Networks in the Humanities; 2022. Available from: https://graphentechnologien.hypotheses.org/files/2022/01/Building_the_Australian_Knowledge_Graph_HuNI_Humanities_Networked_etc-Verhoeven_Burrows.pdf.

[28] Doerr M. The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. In: Kalfoglou Y, Schorlemmer M, Sheth A, Staab S, Uschold M, editors. Semantic Interoperability and Integration. vol. 4391 of Dagstuhl Seminar Proceedings. Leibniz-Zentrum für Informatik; 2005. p. 1-5. doi:10.4230/DagSemProc.04391.22.

[29] Hagen T, Ketzan E. Introducing Traveling Word Pairs in Historical Semantic Change: A Case Study of Privacy Words in 18th and 19th Century English. In: Šeļa A, Jannidis F, Romanowska I, editors. Proceedings of the Computational Humanities Research Conference 2023 (CHR2023). vol. 1613 of CEUR Workshop Proceedings; 2023. p. 461-74. Available from: https://ceur-ws.org/Vol-3558/paper5360.pdf.

[30] Ehrmanntraut A, Hagen T, Konle L, Jannidis F. Type-and Token-based Word Embeddings in the Digital Humanities. In: Ehrmann M, Karsdorp F, Wevers M, Andrews TL, Burghardt M, Kestemont M, et al., editors. Proceedings of the Computational Humanities Research Conference 2023 (CHR2023). vol. 2989 of CEUR Workshop Proceedings. CEUR; 2021. p. 16-38. Available from: https://ceur-ws.org/Vol-2989/long_paper35.pdf.

[31] Hagen T, Ketzan E, Jannidis F, Witt A. Twenty-two Historical Encyclopedias Encoded in TEI: a New Resource for the Digital Humanities. In: DeGaetano S, Kazantseva A, Reiter N, Szpakowicz S, editors. Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2020). International Committee on Computational Linguistics; 2020. p. 112-20. Available from: https://aclanthology.org/2020.latechclfl-1.13.

[32] Matuschek M. Word Sense Alignment of Lexical Resources; 2015. PhD Thesis, Technische Universität Darmstadt. Available from: https://tuprints.ulb.tu-darmstadt.de/id/eprint/4355.

[33] Pais V, Tufiș D, Ion R. MWSA Task at GlobaLex 2020: RACAI's Word Sense Alignment System using a Similarity Measurement of Dictionary Definitions. In: Kernerman I, Krek S, McCrae JP, Gracia J, Ahmadi S, Kabashi B, editors. Proceedings of the 2020 Globalex Workshop on Linked Lexicography. European Language Resources Association; 2020. p. 69-75. Available from: https://aclanthology.org/2020.globalex-1.12.

[34] Loureiro D, Rezaee K, Pilehvar MT, Camacho-Collados J. Analysis and Evaluation of Language Models for Word Sense Disambiguation. Computational Linguistics. 2021;47(2):387-443. doi:10.1162/coli_a_00405.

[35] Wang K, Reimers N, Gurevych I. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. In: Moens MF, Huang X, Specia L, Yih SW, editors. Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics; 2021. p. 671-88. Available from: https://aclanthology.org/2021.findings-emnlp.59.

[36] Hagen T, Jannidis F, Witt A. Word Sense Alignment and Disambiguation for Historical Encyclopedias. In: Andrews T, Diehr F, Efer T, Kuczera A, van Zundert J, editors. Proceedings of the 6th International Conference on Graphs and Networks in the Humanities. Leibniz-Institut für Deutsche Sprache (IDS); 2022. Available from: https://nbn-resolving.org/urn:nbn:de:bsz:mh39-109834.

[37] Ehrmanntraut A. Historical German Text Normalization Using Type- and Token-Based Language Modeling; 2024. ArXiv preprint. doi:10.48550/arXiv.2409.02841.

[38] Hiebel G, Doerr M, Eide Ø. CRMgeo: A spatiotemporal extension of CIDOC-CRM. International Journal on Digital Libraries. 2017;18:271-9. doi:10.1007/s00799-016-0192-4.

[39] Loureiro D, Rezaee K, Pilehvar MT, Camacho-Collados J. Language models and word sense disambiguation: An overview and analysis; 2020. ArXiv preprint. doi:10.48550/arXiv.2008.11608.

[40] Perri V, Qarkaxhija L, Zehe A, Hotho A, Scholtes I. One Graph to Rule them All: Using NLP and Graph Neural Networks to analyse Tolkien's Legendarium. In: Karsdorp F, Nielbo KL, editors. Proceedings of the Computational Humanities Research Conference 2022 (CHR 2022). vol. 3290 of CEUR Workshop Proceedings; 2022. p. 291-317. Available from: `https://ceur-ws.org/Vol-3290/long_paper2171.pdf`.

[41] Gupta T, Hatzel HO, Biemann C. Coreference in Long Documents using Hierarchical Entity Merging. In: Bizzoni Y, Degaetano-Ortlieb S, Kazantseva A, Szpakowicz S, editors. Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024). Association for Computational Linguistics; 2024. p. 11-7. Available from: `https://aclanthology.org/2024.latechclfl-1.2`.

[42] Maaten Lvd, Hinton G. Visualizing Data using t-SNE. Journal of Machine Learning Research. 2008;9(86):2579-605. Available from: `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[43] Zehe A, Konle L, Dümpelmann LK, Gius E, Hotho A, Jannidis F, et al. Detecting scenes in fiction: A new segmentation task. In: Merlo P, Tiedemann J, Tsarfaty R, editors. Proceedings of the 16th conference of the European chapter of the Association for Computational Linguistics (EACL 2021). Main volume. Association for Computational Linguistics; 2021. p. 3167-77. doi:10.18653/v1/2021.eacl-main.276.

[44] Gius E, Jannidis F, Krug M, Zehe A, Hotho A, Puppe F, et al. Detection of Scenes in Fiction. In: Book of Abstracts of the Digital Humanities 2019 (DH2019); 2019. doi:10.34894/OOY9CE.

[45] Zehe A, Konle L, Guhr S, Dümpelmann L, Gius E, Hotho A, et al. Shared Task on Scene Segmentation @ KONVENS 2021. In: Proceedings of the Shared Task on Scene Segmentation. vol. 3001 of CEUR Workshop Proceedings; 2021. p. 1-21. Available from: `http://ceur-ws.org/Vol-3001/paper1.pdf`.

[46] Kurfalı M, Wirén M. Breaking the narrative: Scene segmentation through sequential sentence classification. In: Zehe A, Konle L, Guhr S, Dümpelmann L, Gius E, Hotho A, et al., editors. Proceedings of the Shared Task on Scene Segmentation. vol. 3001; 2021. p. 49-53. Available from: `http://ceur-ws.org/Vol-3001/paper6.pdf`.

[47] Ehrmanntraut A, Konle L, Jannidis F. LLpro: A Literary Language Processing Pipeline for German Narrative Texts. In: Georges M, Herygers A, Friedrich A, Roth B, editors. Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023). Association for Computational Linguistics; 2023. p. 28-39. Available from: `https://aclanthology.org/2023.konvens-main.3`.

[48] Konle L, Jannidis F. Modeling Plots of Narrative Texts as Temporal Graphs. In: Karsdorp F, Nielbo KL, editors. Proceedings of the Computational Humanities Research Conference 2022 (CHR 2022). CEUR Workshop Proceedings. CEUR; 2022. p. 318-36. Available from: `https://ceur-ws.org/Vol-3290/long_paper2313.pdf`.

[49] Rozemberczki B, Scherer P, He Y, Panagopoulos G, Riedel A, Astefanoaei M, et al. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. Association for Computing Machinery; 2021. p. 4564–4573. doi:10.1145/3459637.3482014.

[50] Zhang R, Reddy RG, Sultan MA, Castelli V, Ferritto A, Florian R, et al. Multi-Stage Pre-training for Low-Resource Domain Adaptation. In: Webber B, Cohn T, He Y, Liu Y, editors. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics; 2020. p. 5461-8. Available from: `https://aclanthology.org/2020.emnlp-main.440/`.

[51] Gururangan S, Marasović A, Swayamdipta S, Lo K, Beltagy I, Downey D, et al. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2020. p. 8342-60. doi:10.18653/v1/2020.acl-main.740.

[52] Diao S, Xu T, Xu R, Wang J, Zhang T. Mixture-of-Domain-Adapters: Decoupling and Injecting Domain Knowledge to Pre-trained Language Models' Memories. In: Rogers A, Boyd-Graber J, Okazaki

N, editors. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2023. p. 5113-29. Available from: https://aclanthology.org/2023.acl-long.280.

[53] Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: Enhanced Language Representation with Informative Entities. In: Korhonen A, Traum D, Màrquez L, editors. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2019. p. 1441-51. doi:10.18653/v1/P19-1139.

[54] Wang X, Gao T, Zhu Z, Zhang Z, Liu Z, Li J, et al. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. Transactions of the Association for Computational Linguistics. 2021;9:176-94. doi:10.1162/tacl_a_00360.

[55] Moiseev F, Dong Z, Alfonseca E, Jaggi M. SKILL: Structured Knowledge Infusion for Large Language Models. In: Carpuat M, de Marneffe MC, Meza Ruiz IV, editors. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2022. p. 1581-8. doi:10.18653/v1/2022.naacl-main.113.

[56] Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, et al. K-bert: Enabling language representation with knowledge graph. In: Proceedings of the AAAI Conference on Artificial Intelligence: Technical Tracks 3. AAAI Conference and Symposium Proceedings. Association for the Advancement of Artificial Intelligence; 2020. p. 2901-8. doi:10.1609/aaai.v34i03.5681.

[57] Beltagy I, Lo K, Cohan A. SciBERT: A pretrained language model for scientific text; 2019. ArXiv preprint. doi:10.48550/arXiv.1903.10676.

[58] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding; 2018. ArXiv preprint. doi:10.48550/arXiv.1810.04805.

[59] Fichtel L, Kalo JC, Balke WT. Prompt tuning or fine-tuning-investigating relational knowledge in pre-trained language models. In: Proceedings of the 3rd Conference on Automated Knowledge Base Construction (AKBC 2021); 2021. Available from: https://openreview.net/forum?id=o7sMlpr9yBW.

[60] Yao L, Mao C, Luo Y. KG-BERT: BERT for knowledge graph completion; 2019. ArXiv preprint. doi:10.48550/arXiv.1909.03193.

[61] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter; 2019. ArXiv preprint. doi:10.48550/arXiv.1910.01108.

[62] Clark K, Luong MT, Le QV, Manning CD. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators; 2020. ArXiv preprint. doi:10.48550/arXiv.2003.10555.

[63] Konle L, Jannidis F. Domain and Task Adaptive Pretraining for Language Models. In: Karsdorp F, McGillivray B, Nerghes A, Wevers M, editors. Proceedings of the Workshop on Computational Humanities Research (CHR 2020). vol. 2723 of CEUR Workshop Proceedings; 2020. p. 248-56. Available from: https://ceur-ws.org/Vol-2723/short33.pdf.

[64] Krug M, Weimer L, Reger I, Macharowsky L, Feldhaus S, Puppe F, et al.. Description of a Corpus of Character References in German Novels: DROC [Deutsches ROman Corpus]; 2017. Nr. 27 in DARIAH-DE Working Papers. Available from: https://resolver.sub.uni-goettingen.de/purl?gro-2/108301.

[65] Brunner A, Engelberg S, Jannidis F, Tu NDT, Weimer L. Corpus REDEWIEDERGABE. In: Calzolari N, Béchet F, Blache P, Choukri K, Cieri C, Declerck T, et al., editors. Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020). European Language Resources Association; 2020. p. 803-12. Available from: https://www.aclweb.org/anthology/2020.lrec-1.100.

[66] Speer R, Chin J, Havasi C. ConceptNet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017). AAAI Press; 2017. p. 4444-51. doi:10.1609/aaai.v31i1.11164.

[67] Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA. Retrofitting Word Vectors to Semantic Lexicons. In: Mihalcea R, Chai J, Sarkar A, editors. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2015. p. 1606-15. doi:10.3115/v1/N15-1184.

[68] Lengerich B, Maas A, Potts C. Retrofitting Distributional Embeddings to Knowledge Graphs with Functional Relations. In: Bender EM, Derczynski L, Isabelle P, editors. Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics; 2018. p. 2423-36. Available from: https://aclanthology.org/C18-1205.

[69] Thoma S, Rettinger A, Both F. Towards holistic concept representations: Embedding relational knowledge, visual attributes, and distributional word semantics. In: d'Amato C, Fernandez M, Tamma V, Lecue F, Cudré-Mauroux P, Sequeda J, et al., editors. Proceedings of the 16th International Semantic Web Conference (IWSC 2017). No. 10587 in Lecture Notes in Computer Science. Springer; 2017. p. 694-710. doi:10.1007/978-3-319-68288-4_41.

[70] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating Embeddings for Modeling Multi-relational Data. In: Burges CJ, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems (NIPS 2013). vol. 26. Curran Associates; 2013. Available from: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Abstract.pdf.

[71] Sun Z, Deng ZH, Nie JY, Tang J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In: The Seventh International Conference on Learning Representations (ICLR 2019); 2019. Available from: https://openreview.net/pdf?id=HkgEQnRqYQ.

[72] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space; 2013. ArXiv preprint. doi:10.48550/arXiv.1301.3781.

[73] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 2017;5:135-46. doi:10.1162/tacl_a_00051.

[74] Ortiz Suárez PJ, Romary L, Sagot B. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In: Jurafsky D, Chai J, Schluter N, Tetreault J, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2020. p. 1703-14. doi:10.18653/v1/2020.acl-main.156.

[75] Hamp B, Feldweg H. GermaNet. A Lexical-Semantic Net for German. In: Vossen P, Adriaens G, Calzolari N, Sanfilippo A, Wilks Y, editors. Proceeedings of the ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. Association for Computational Linguistics; 1997. Available from: https://aclanthology.org/W97-0802.

[76] Hagen T. Verwendung von Wissensgraphen zur inhaltlichen Ergänzung kleinerer Textkorpora. In: Geierhos M, editor. Abstracts zur 8. Jahrestagung des Verbands Digital Humanities im deutschsprachigen Raum e.V. (DHd2022); 2022. doi:10.5281/zenodo.6328009.

[77] Köper M, Scheible C, Schulte im Walde S. Multilingual Reliability and "Semantic" Structure of Continuous Word Spaces. In: Purver M, Sadrzadeh M, Stone M, editors. Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015). Association for Computational Linguistics; 2015. p. 40-5. Available from: https://aclanthology.org/W15-0105.

[78] Finkelstein L, Gabrilovich E, Matias Y, Rivlin E, Solan Z, Wolfman G, et al. Placing search in context: The concept revisited. In: Proceedings of the 10th International Conference on World Wide Web (WWW10); 2001. p. 406-14. Available from: https://www.ra.ethz.ch/CDstore/www10/papers/pdf/p431.pdf.

[79] Leviant I, Reichart R. Separated by an un-common language: Towards judgment language informed vector space modeling; 2015. ArXiv preprint. doi:10.48550/arXiv.1508.00106.

[80] Hill F, Reichart R, Korhonen A. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. Computational Linguistics. 2015;41(4):665-95. doi:10.1162/COLI_a_00237.

[81] Zesch T, Gurevych I, Mühlhäuser M. Comparing Wikipedia and German Wordnet by Evaluating Semantic Relatedness on Multiple Datasets. In: Sidner C, Schultz T, Stone M, Zhai C, editors. Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics; 2007. p. 205-8. Available from: https://aclanthology.org/N07-2052.

[82] Rubenstein H, Goodenough JB. Contextual correlates of synonymy. Communications of the ACM. 1965;8(10):627-33. doi:10.1145/365628.365657.

[83] Gurevych I. Using the Structure of a Conceptual Network in Computing Semantic Relatedness. In: Dale R, Wong KF, Su J, Kwong OY, editors. Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP 2005). No. 3651 in Lecture Notes in Computer Science. Springer; 2005. doi:10.1007/11562214_67.

[84] Omeliyanenko J, Zehe A, Hettinger L, Hotho A. LM4KG: Improving common sense knowledge graphs with language models. In: Pan JZ, Tamma V, d'Amato C, Janowicz K, Fu B, Polleres A, et al., editors. Proceedings of the 19th International Semantic Web Conference (IWSC 2020). No. 12506 in Lecture Notes in Computer Science. Springer; 2020. p. 456-73. doi:10.1007/978-3-030-62419-4_26.

[85] Petroni F, Rocktäschel T, Riedel S, Lewis P, Bakhtin A, Wu Y, et al. Language Models as Knowledge Bases? In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). Association for Computational Linguistics; 2019. p. 2463-73. doi:10.18653/v1/D19-1250.

[86] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners; 2019. Available from: `https://openai.com/index/better-language-models/`.

[87] Chen X, Liu X, Ragni A, Wang Y, Gales MJ. Future word contexts in neural network language models. In: Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE; 2017. p. 97-103. doi:10.1109/ASRU.2017.8268922.

[88] Bruni E, Tran NK, Baroni M. Multimodal distributional semantics. Journal of Artificial Intelligence Research. 2014;49:1-47. doi:10.1613/jair.4135.

[89] Luong T, Socher R, Manning C. Better Word Representations with Recursive Neural Networks for Morphology. In: Hockenmaier J, Riedel S, editors. Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL 2013). Association for Computational Linguistics; 2013. p. 104-13. Available from: `https://aclanthology.org/W13-3512`.

[90] Halawi G, Dror G, Gabrilovich E, Koren Y. Large-scale learning of word relatedness with constraints. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012). Association for Computing Machinery; 2012. p. 1406-14. doi:10.1145/2339530.2339751.

[91] Camacho-Collados J, Pilehvar MT, Collier N, Navigli R. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In: Bethard S, Carpuat M, Apidianaki M, Mohammad SM, Cer D, Jurgens D, editors. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics; 2017. p. 15-26. doi:10.18653/v1/S17-2002.

[92] Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences. 2017;114(13):3521-6. doi:10.1073/pnas.1611835114.

[93] Omeliyanenko J, Zehe A, Hotho A, Schlör D. CapsKG: Enabling Continual Knowledge Integration in Language Models for Automatic Knowledge Graph Completion. In: Payne TR, Presutti V, Qi G, Poveda-Villalón M, Stoilos G, Hollink L, et al., editors. Proceedings of the 22th International Semantic Web Conference (ISWC 2023). No. 14266 in Lecture Notes in Computer Science. Springer Nature; 2023. p. 618-36. doi:10.1007/978-3-031-47240-4_33.

[94] Bosselut A, Rashkin H, Sap M, Malaviya C, Celikyilmaz A, Choi Y. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In: Korhonen A, Traum D, Màrquez L, editors. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2019. p. 4762-79. doi:10.18653/v1/P19-1470.

[95] Agarwal O, Ge H, Shakeri S, Al-Rfou R. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In: Toutanova K, Rumshisky A, Zettlemoyer L, Hakkani-Tur D, Beltagy I, Bethard S, et al., editors. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2021. p. 3554-65. doi:10.18653/v1/2021.naacl-main.278.

[96] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, et al. Parameter-Efficient Transfer Learning for NLP. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research; 2019. p. 2790-9. Available from: `http://proceedings.mlr.press/v97/houlsby19a/houlsby19a.pdf`.

[97] Kumar A, Singh SS, Singh K, Biswas B. Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications. 2020;553. Article 124289. doi:10.1016/j.physa.2020.124289.

[98] Mahdisoltani F, Biega J, Suchanek FM. YAGO3: A Knowledge Base from Multilingual Wikipedias. In: Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR 2015); 2015. Available from: `http://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf`.

[99] Tran HN, Takasu A. Raedt LD, editor. MEIM: Multi-partition Embedding Interaction Beyond Block Term Format for Efficient and Expressive Link Prediction. International Joint Conferences on Artificial Intelligence; 2022. doi:10.24963/ijcai.2022/314.

[100] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference. In: Allauzen A, Grefenstette E, Hermann KM, Larochelle H, Yih SWt, editors. Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. Association for Computational Linguistics; 2015. p. 57-66. doi:10.18653/v1/W15-4007.

[101] Lauscher A, Majewska O, Ribeiro LF, Gurevych I, Rozanov N, Glavaš G. Common Sense or World Knowledge? Investigating adapter-based Knowledge Injection into Pretrained Transformers; 2020. ArXiv preprint. doi:10.48550/arXiv.2005.11787.

[102] Mollo DC, Millière R. The vector grounding problem; 2023. ArXiv preprint. doi:10.48550/arXiv.2304.01481.