# A New Time-sensitive Model of Linguistic Knowledge for Graph Databases

Pierpaolo Basile[1], Pierluigi Cassotti[1], Stefano Ferilli[1,*] and Barbara McGillivray[2]

[1]University of Bari Aldo Moro, Department of Computer Science, Via E. Orabona 4, Bari, 70125, Italy
[2]King's College London, Strand Campus, Strand, London, WC2R 2LS, United Kingdom

**Abstract**

Graph databases are a straightforward technology for storing knowledge graphs. However, they are schema-less. We apply the GraphBRAIN Schema (GBS) format to describe Time-sensitive Linguistic Knowledge in a graph database (Neo4j). Our schema can model relations between concepts and words, information about word occurrences, and diachronic information about concepts and words. This paper introduces GraphBRAIN technology and describes our model for time-sensitive linguistic data. Moreover, we provide an example of usage and show the potential of this model for humanities and cultural heritage research.

**Keywords**

Knowledge Graphs, Linguistic Resources, Graph Databases, Graph Data Model, Diachronic Language Data

## 1. Introduction and Motivation

ICT provides an unprecedented opportunity to foster and support the preservation and research on immaterial Cultural Heritage. A large part of research in the Humanities and Cultural Heritage (H&CH) sector involves the collection and analysis of the material of cultural and/or historical interest. Semantic Web technologies have been used successfully in a number of humanities projects such as the Pelagios project [1] and the Mapping the Manuscripts project [2]. Given the relevance of textual materials in this research, it is not surprising that significant progress has been made in the design of linked data models for language data (see, for example, the excellent survey in Khan et al. [3]). A notable example of a multilingual synchronic language resource that has had a profound impact on the research community is BabelNet [4], a semantic network which connects the English computational lexicon WordNet [**?** ] with a range of Open Linked Data resources such as Wikipedia and Wikidata, and many others. Alongside such resources, the research community has developed Semantic Web ontologies such as LeMON [5] particularly designed for the encoding of linguistic information.

*1st Italian Workshop on Artificial Intelligence for Cultural Heritage (AI4CH22), co-located with the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022). 28 November 2022, Udine, Italy.*

*Corresponding author.
pierpaolo.basile@uniba.it (P. Basile); pierluigi.cassotti@uniba.it (P. Cassotti); stefano.ferilli@uniba.it (S. Ferilli); barbara.mcgillivray@kcl.ac.uk (B. McGillivray)
0000-0002-0545-1105 (P. Basile); 0000-0001-7824-7167 (P. Cassotti); 0000-0003-1118-0601 (S. Ferilli); 0000-0003-3426-8200 (B. McGillivray)

The ability to model (language) data *diachronically*, is particularly important as a large part of HCH work deals with historical data with a view to model change over time. In this line of research, some work has started on the modelling of cognate words and loan relations between words [6]. Related to this is the treatment of semantic change, the phenomenon concerned with the change in the meaning of words over time. The automatic detection of such changes has seen a very rapid development in Natural Language Processing (NLP) research in recent years [7, 8, 9], with the majority of the approaches relying on distributional semantics, i.e. on representations of the semantics of words trained from corpus data covering different time intervals via embedding technologies. Some studies, e.g. [10], have advocated for the integration of such distributional approaches with Linked Open Data technologies, stressing how this best caters for the heterogeneous nature of the data relevant to this phenomenon, which includes not only language data, but also data on historical events and entities, as well as of bibliographic and geographic nature. However, Linked Open Data technologies have some limitations which we propose to address in this paper, as explained below.

Data Bases (DBs) aim at efficient storage, management and retrieval of data. Knowledge Bases (KBs), investigated in AI, are aimed at supporting formal reasoning on the available information. A *Knowledge Graph* (KG) is a kind of KB [11] where an ontology acts as the data model, and the data are organized in a graph structure [12]:

$$\text{ontology} + \text{data} = \text{knowledge graph}.$$

Combining the advantages of Database Management Systems (DBMSs) for handling individuals (scalability, storage optimization, efficient handling, mining and browsing of the data, etc.) with the high-level functionalities available in KBs would endow applications with much more power than allowed by the DB's query language alone.

An opportunity for such combination comes from the recent development of *Graph Databases*, a kind of NoSQL DBs of which Neo4j [13] is the most popular representative. Neo4j has been adopted by many big companies and governmental organizations for several different and relevant use cases, including Recommendation, Biology, Artificial Intelligence and Data Analytics, Social Networks, Data Science and Knowledge Graphs[1]. Neo4j comes with a powerful query language (Cypher) and extensive libraries for advanced data manipulation (APOC).

Unfortunately, formal ontologies and graph DBs refer to different graph models, which cannot straightforwardly be combined together. The standard formalism for expressing ontologies and KGs is the Web Ontology Language (OWL)[2], based on the Resource Definition Framework (RDF)[3]. RDF graphs are built upon RDF Triples of the form:

$$(\text{Subject}, \text{Predicate}, \text{Object})$$

representing arcs between the Subject and Object nodes. A more general structure is provided by the Labeled Property Graphs (LPGs) model [14] (adopted by Neo4j), ensuring great flexibility and expressive power. In LPGs, both nodes and arcs are associated with unique identifiers,

[1]https://neo4j.com/use-cases/, consulted September 8, 2021.
[2]https://www.w3.org/OWL/
[3]https://www.w3.org/RDF/

may be labeled, and can store *properties* represented as key/value maps. Relevant advantages brought by LPGs over RDF graphs are[4]:

- In RDF graphs nodes are atomic, while in LPGs they carry information; this ensures a much more compact structure in the latter. Consequently, RDF graphs are much less readable and they also cause a significant decay in efficiency, especially in browsing-intensive tasks such as Social Network Analysis or Graph Mining algorithms;
- RDF cannot distinguish different occurrences of the same relationship between the same pair of entities; this is possible in LPGs thanks to the unique identifiers of relationships instances;
- RDF cannot attach properties to instances of relationships; the reification solution (transforming a relationship instance into an object which has relationships to the original Subject and Object and to the additional properties) worsens readability; another partial solution is via annotations.

One limitation of Neo4j is that it is schema-less: the user may apply any label/type or property to each single node or arc. While ensuring great flexibility, this means that there is no clear semantics for the graph contents. Developing LPG-based KGs requires the definition of an LPG-based ontological formalism for expressing graph DB schemas, so as to allow data interpretability and applications interoperability, and of a mapping between this model and the standard ontological model adopted in the literature. Research on this topic resulted in the *GraphBRAIN* technology, whose peculiarities and advantages are discussed in [15]. In GraphBRAIN the KB designers must provide pre-specified data schemas, expressed in the form of LPG-based ontologies, that will drive all subsequent accesses to a Neo4j graph DB. By referring to a schema, the applications will commit to be compliant with it, as in traditional databases. In this work, we will adopt GraphBRAIN technology to model time-sensitive linguistic knowledge in a graph database.

## 2. GraphBRAIN Graph Database Scheme Format

The *GraphBRAIN Schema* (GBS) format to define graph DB schemas consists of an XML file whose tags allow us to exploit the representational features provided for by the LPG model. Here we will recall its main components (more details can be found in [15]).

The main structure of the XML tags is reported in Listing 1, where the universal entity *Entity* and the universal relationship *Relationship*, acting respectively as the roots of the entity and relationship hierarchies, are implicitly assumed (recall that in ontological terminology entities correspond to classes and relationships correspond to object properties). Therefore, entities and relationships are to be specified only starting from the first level of specialization, which we will call *top level*. Since each node (resp., arc) in the graph must be associated with one top-level entity (resp., relationship), the top-level entities (resp., relationships) are to be considered as disjoint. They may be the roots of specialization hierarchies of sub-entities (resp., sub-relationships). The set of direct specializations of a (sub-)entity or (sub-)relationship are in turn disjoint and are not to be intended as a partition: instances that do not fit any of the

[4]https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/, consulted September 8, 2021.

specializations of a parent (sub-)entity or (sub-)relationship may be directly associated with the parent. This design choice prevents multiple inheritances, i.e. associating an instance to many classes belonging to different branches in the hierarchy. We partially recover this at the level of instances: when two instances of different (sub-)entities represent the same object, we link them using an *aliasOf* relationship. The single reference object represented by all these instances takes the union of their attributes.

```
domain // tag enclosing the overall ontology
    [imports]
    entities // tag enclosing the classes
        {entity} // see (*)
    relationships // tag enclosing the relationships
        {relationship} // see (*)
```

Listing 1: Main structure of GBS files.

Entities and relationships are specified using the structure shown in Listing 2. **Reference** is used only in relationships to specify their possible domain-range pairs, **taxonomy** allows us to conveniently represent the specialization-type assertions; all other object properties are to be specified in the **relationships** section. **Attributes** are mandatory for entities (an entity instance must be described by some attribute) and optional for relationships (a relationship may carry information in its very linking two instances). **Specialization** is a recursive tag, allowing us to define hierarchies of sub-entities or sub-relationships. In addition to its own attributes, each specialization inherits all the attributes of the (sub-)entities (resp., (sub-)relationships) on the hierarchy path from its specific **specialization** section up to the corresponding top-level entity (resp., relationship).

```
(*) ( entity | relationship | specialization ) tag
    [references]
        {reference}
    [taxonomy]
        {specialization} // see (*) (recursive)
    [attributes] // specifying the data properties
        {attribute}
```

Listing 2: Structure for describing entity and relationship hierarchies in GBS files.

Regarding datatypes, attributes of type *integer*, *real*, *boolean*, *string*, *text* take an atomic value of the corresponding type, where *text* is intended for free text of any length. This is different from *string*, which has a limited maximum length that can be specified in the 'length' attribute. Attributes of type *date* take values in one of the following forms: year; year/month;

year/month/day. Attributes of type *select* denote a choice in an enumeration of values; attributes of type *tree* denote a choice in a tree of values; attributes of type *entity* denote 1:1 relationships between an instance of the current entity and an instance of another entity (specified in the 'target' attribute of the tag), e.g., the birthplace of an entity Person would be modeled as an attribute of type *entity* with target='Place'.

Each GBS schema is intended to describe one domain. However, sometimes wider domains involve ontological elements that are already described in more 'basic' schemas: for example, the schemas for Cultural Heritage, Food and Transportations might be exploited in the ontology aimed at supporting a touristic application. In such cases, it might be useful to reuse such schemas, both to standardize the definitions and to build on existing knowledge. The combination of multiple schemas is more powerful a representation than the simple juxtaposition of their elements. Indeed, their shared entities act as bridges that allow, through the relationships available in those domains, to connect proprietary entities of each domain that would not otherwise have a chance to be related with each other. In the GBS framework, classes and relationships in different ontologies are considered the same (and thus are shared) if they have the same name. They may have, however, different attributes, reflecting the different perspectives associated with the different domains. If an attribute is present in different domains it must have the same type in all of them. Moreover, additional cross-schema relationships (and entities) may be defined in the overall ontology, building on the existing ones. GBS schemas support such scenarios by providing for an optional section in which existing schemas can be imported.

## 3. Mapping onto DB and Ontology

Since graph DBs are naturally suited to express knowledge graphs, i.e., knowledge bases based on given ontologies, a fundamental requirement of our approach is that our schemas can be mapped onto both the DB and to an OWL representation which can then be processed by a reasoner. In this section, we report how these two mappings work in practice.

As said, part of the main motivation for defining GBS schemas is to endow LPG-based graph DBs with a schema that ensures a clear semantics to the information pieces they contain and provides directions for their management and interpretation. In this perspective, the DB users will be required to work according to pre-specified data schemas expressed in the form of ontologies.

In our approach we allow a single graph DB to underlie several domains (schemas), provided that their elements (entities and relationships) are compatible. Each such schema would provide a partial view of the DB contents, perhaps representing a different perspective.

Let us now show how the GBS elements are implemented using LPG features. Leveraging the possibility of using many labels for nodes, each node is labeled with the specific entity it belongs to and with all the domains for which it is relevant (e.g., 'Herbert Simon' would be labeled with 'Person' for the entity and with 'economy' and 'computing' for the domains). On the other hand, since each arc may take at most one type, we use it for specifying the relationship it expresses.

Concerning attributes, a reserved attribute *notes* is implicitly assumed for both nodes and arcs, which allows us to add information not accounted for by the other, domain-specific attributes.

Attribute values of types *integer*, *real*, *boolean*, *string* and *text* are stored as literal values for the corresponding DB types, e.g., Neo4j provides the following types matching GBS types: Integer and Float, Boolean, and String. For types *select* and *tree* the string corresponding to the selected value in the list or tree is stored. An attribute of type *entity* actually corresponds to a relationship between the current instance and an instance of the target entity and thus it is stored in the DB as an arc, connecting the nodes corresponding to these two instances and having the attribute name as type. Finally, albeit Neo4j provides for temporal types, including 'Date', following [13] we propose to model attributes of type *date* as relationships to one of the following four entities: **Day** (representing a specific day of a specific year, with integer attributes *day*, *month*, *year*); **Month** (representing a specific month of a specific year, with integer attributes *month*, *year*); **Year** (representing a year, with a single integer attribute *year*).

## 4. Linguistic Knowledge Graph

This section briefly describes the first version of our Linguistic Knowledge Graph. Figure 1 is a sketch of the Graph Data Model. Our graph aims to model: 1) relations between concepts and words; 2) information about word occurrences; 3) diachronic information of both concepts and words. Moreover, we design the graphs considering further extensions, mainly morphological features and historical events.

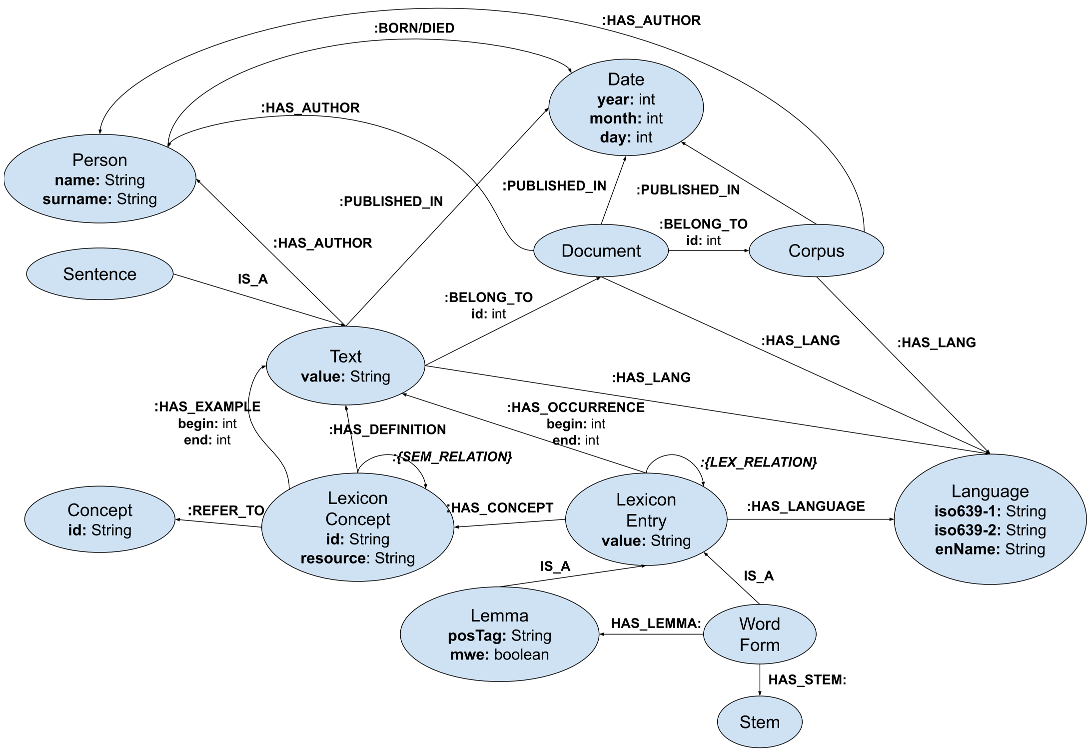


**Figure 1:** Graph Data Model of our Linguistic Knowledge Graph.

Following the structure of existing semantic networks (e.g., WordNet and BabelNet) and ontologies (LeMON), we design our graph starting from the *LexiconConcept* that represents a concept in a specific resource (e.g., WordNet or BabelNet). The *LexiconConcept* is linked to a *Concept*. This structure allows us to define the same concept in different resources. The lexicon concepts are connected through several semantic relations (hyperonym, hyponym, etc.), generally represented by the edge *:{SEM_RELATION}* in the model. Several lexicon entries can identify the same *LexiconConcept*. A *LexiconEntry* is a surface form that refers to one or more concepts. Lexicon entries can be connected by lexical relations (*:{LEX_RELATION}* in the graph model).

*LexiconEntry* is the root of a hierarchy that includes *Lemma* and *WordForm*. A *WordForm* can have a *Stem*. The Lemma class has attributes such as a part-of-speech tag (*posTag*) and a boolean (*mwe*) indicating if the lemma is a multi-word expression or not (e.g., *Artificial Intelligence*). Each *LexiconEntry* refers to a specific *Language*.

The class *Text* represents any passage of text, and its hierarchy includes *Sentence*. In particular, the relation *:HAS_EXAMPLE* indicates a passage of text that contains an example of the usage of a specific *LexiconConcept*. *:HAS_EXAMPLE* has the attributes *begin* and *end* that indicate the character offset of where the example starts and ends, respectively. Each *Text* can belong to a specific *Document*, and a *Document* can be part of a *Corpus*. The *:BELONG_TO* relation has an id that univocally identifies a text in a document and a document in a corpus.

Concerning the diachronic aspect, we model the concept of *Date* that can be linked to *Text*, *Document*, and *Corpus*. For example, using this structure we can express when a concept was used. The graph model also reports *Document*, and *Corpus* and the class *Person*, which can be linked to *Text* by the *:HAS_AUTHOR* relation. Moreover, the class *Author* is linked to *Date* through the relations *:BORN* and *:DIED*. The *Author* is an example of a class external to the linguistic domain and it highlights how we can easily add classes to our model. In the future, we plan to include classes for modeling events and linking them to *Date* and *Author*.

Figure 2 shows a fragment of the XML file expressing the data model as an LPG ontology, focusing on *LexiconConcept* and *LexiconEntry* entities. In our ontology they are both subclasses of a *ContentDescription* entity, and they both have subclasses. In particular, subclasses of *LexiconEntry* are *Lemma* and *WordForm*. Some subclasses add their own attributes to those inherited by superclasses. Some attributes are required, and some are optional. Some relevant relationships among these entities are *LexiconEntry.hasConcept.LexiconConcept*, *WordForm.hasLemma.Lemma* and *WordForm.hasStem.Stem*.

## 5. Use case

Figure 3 shows the sub-graph related to the Lexicon Entry *plane*. The extracted sub-graph shows the concepts associated with the Lexicon Entry by the referring lexical resource, in this case WordNet, using the HAS_CONCEPT relationship. The concepts sketched are the synsets *airplane.n.01*, *plane.n.02*, *plane.n.03*, *plane.n.04*, *plane.v.01*, *plane.v.02*. For each Lexicon Concept, the WordNet glosses are provided by the relation *HAS_DEFINITION*.

Moreover, the example sub-graph includes a sentence extracted by the book *The Last Enemy* and containing the word *plane*, i.e.

```
<entities>
  <entity name=”ContentDescription”>
    <attributes>
      <attribute name=”name” mandatory=”true” datatype=”string”/>
      <attribute name=”description” mandatory=”false” datatype=”string”/>
    </attributes>
    <taxonomy>
      <value name=”LexiconConcept”>
        <attributes>
          <attribute name=”id” mandatory=”true” datatype=”string”/>
          <attribute name=”resource” mandatory=”true” datatype=”string”/>
          <attribute name=”taxonomy” mandatory=”false” datatype=”entity” target=”Taxonomy” />
        </attributes>
        <taxonomy>
          ...
        </taxonomy>
      </value>
      <value name=”LexiconEntry”>
        <attributes>
          <attribute name=”pos” mandatory=”true” datatype=”string”/>
          <attribute name=”language” mandatory=”false” datatype=”entity” target=”language” />
        </attributes>
        <taxonomy>
          <value name=”Lemma”>
            <attributes>
              <attribute name=”posTag” mandatory=”true” datatype=”string”/>
              <attribute name=”mwe” mandatory=”false” datatype=”boolean” />
            </attributes>
          </value>
          <value name=”WordForm”>
            <taxonomy>
              <value name=”Stem”>
            </taxonomy>
          </value>
        </taxonomy>
      </value>
    </taxonomy>
  </entity>
  ...
<entities>
<relationships>
  <relationship name=”hasConcept” inverse=”conceptOf”>
    <references>
      <reference subject=”LexiconEntry” object=”LexiconConcept”/>
    </references>
  </relationship>
  <relationship name=”hasLemma” inverse=”lemmaOf”>
    <references>
      <reference subject=”WordForm” object=”Lemma”/>
    </references>
  </relationship>
  <relationship name=”hasStem” inverse=”stemOf”>
    <references>
      <reference subject=”WordForm” object=”Stem”/>
    </references>
  </relationship>
  ...
</relationships>
```

**Figure 2:** Fragment of the schema expressed in LPG ontology format

*“My plane had been fitted out with a new cockpit hood”.*

The book is represented as a Document instance and belongs to the corpus Gutenberg (rel. BELONG_TO). The rel. *HAS_AUTHOR* connects the book with the author *Richard Hillary*, who was born on the 20th of April 1919 (rel. *BORN*) and died on the 08th of January 1943 (rel. *DIED*). The book publishing date, i.e. 1942, can be obtained via the rel. *PUBLISHED_IN*.

The extracted sentence is connected to both the Lexicon Entry *plane* and the Lexicon Concept *airplane.n.01* respectively by the rels. *HAS_OCCURRENCE* and *HAS_EXAMPLE*. The former rel. addresses the occurrence of the word *plane* in the sentence, the latter that *plane* occurs with the meaning specified by the Lexicon Concept *airplane.n.01*, i.e.

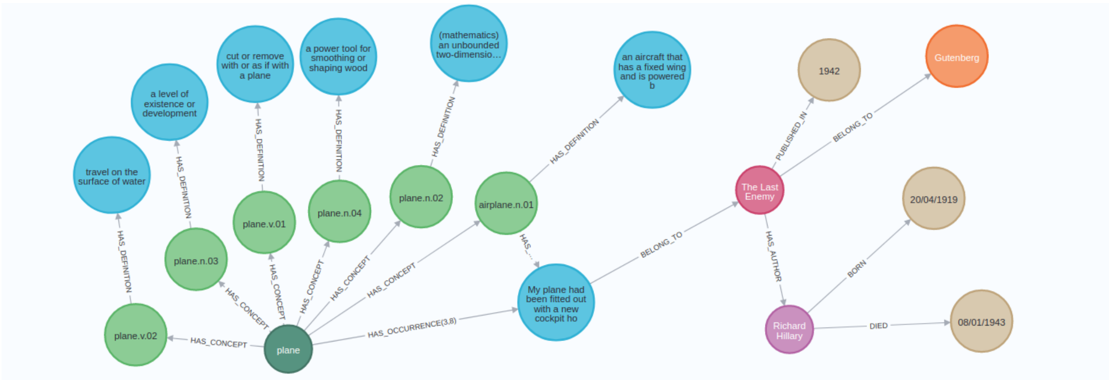


**Figure 3:** Example of a sub-graph for the Lexicon Entry *plane*.

*"an aircraft that has a fixed wing and is powered by propellers or jets".*

In both relations, the offsets of the word *plane* are specified by the relationship attributes, i.e. 3 and 8.

In the proposed example, the time dimension is elicited by three components: the book publishing date, the date of birth and the date of death of the author *Richard Hillary*. The time specifications acquire a relevant role in the context of Diachronic Linguistics. From the publishing date of *The Last Enemy*, we can infer that the occurrence of *plane* in the extracted sentence is one of the earlier appearances of the word *plane* with the *airplane.n.01* sense. Furthermore, information about the Author, such as his influences, and the historical period in which he lived, can enable deeper analyses, guiding the study and the definition of the cause-effects relationship in Lexical Semantic Change phenomena.

## 6. Conclusions and Future Work

We have presented a new time-sensitive model of linguistic knowledge that can be used for graph databases. This model has the potential to be employed in a range of applications in the CH sector and in Humanities research. In linguistics, the model can be used to investigate word histories and conduct etymological research, as well as the analysis of quantitative patterns in the distribution of word senses not only across time (semantic change), but also according to the authors of the texts and other textual features (semantic variation). Moreover, the model's ability to connect word meaning instances in texts with lexical concepts also enables applications in the growing area of word sense disambiguation from historical texts, which aims to associate the most likely meaning of a word to an instance of usage of that word in a historical text [16, 17]. More broadly, research in representing word meaning change in texts can be fruitfully applied to the automatic detection of concept drift, i.e. the phenomenon by which the definition, properties, instances or label of a concept change [18, 19], has been shown to have relevant implications for conceptual history research [20].

In future work, we plan to extend our model to include more information about events. Moreover, we will import data from several semantic networks and linked open data.

## References


[1] L. Isaksen, R. Simon, E. T. Barker, P. de Soto Cañamares, Pelagios and the emerging graph of ancient world data, in: Proceedings of the 2014 ACM conference on Web science, 2014, pp. 197–201.

[2] T. Burrows, E. Hyvönen, L. Ransom, H. Wijsman, Mapping manuscript migrations: Digging into data for the history and provenance of medieval and renaissance manuscripts, Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies 3 (2018) 249–252.

[3] A. F. Khan, C. Chiarcos, T. Declerck, D. Gifu, E. G.-B. García, J. Gracia, M. Ionov, P. Labropoulou, F. Mambrini, J. P. McCrae, E. Pagé-Perron, M. Passarotti, S. R. Muñoz, C.-O. Truică, When linguistics meets web technologies. recent advances in modelling linguistic linked data, Semantic Web (2022) 1–64. doi:10.3233/SW-222859.

[4] R. Navigli, S. Ponzetto, Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artificial Intelligence (2012) 217–250.

[5] J. P. McCrae, D. Spohr, P. Cimiano, Linking lexical resources and ontologies on the semantic web with lemon, in: G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, J. Z. Pan (Eds.), The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I, volume 6643 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 245–259. URL: https://doi.org/10.1007/978-3-642-21034-1_17. doi:10.1007/978-3-642-21034-1_17.

[6] F. Abromeit, C. Chiarcos, C. Fäth, M. Ionov, Linking the Tower of Babel: Modelling a Massive Set of Etymological Dictionaries as RDF, in: Proceedings of the 5th Workshop on Linked Data in Linguistics (LDL-2016): Managing, Building and Using Linked Language Resources, 2016, pp. 11–19.

[7] N. Tahmasebi, L. Borin, A. Jatowt, Survey of computational approaches to lexical semantic change, arXiv: Computation and Language (2018).

[8] A. Tsakalidis, M. Bazzi, M. Cucuringu, P. Basile, B. McGillivray, Mining the UK web archive for semantic change detection, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2019, Varna, Bulgaria, September 2-4, 2019, INCOMA Ltd., 2019, pp. 1212–1221. URL: https://doi.org/10.26615/978-954-452-056-4_139. doi:10.26615/978-954-452-056-4_139.

[9] D. Schlechtweg, B. McGillivray, S. Hengchen, H. Dubossarsky, N. Tahmasebi, SemEval-2020 task 1: Unsupervised lexical semantic change detection, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 1–23. URL: https://aclanthology.org/2020.semeval-1.1. doi:10.18653/v1/2020.semeval-1.1.

[10] Armaselu, Florentina and Apostol, Elena-Simona and Khan, Anas Fahad and Liebeskind, Chaya and McGillivray, Barbara, and Truică, Ciprian-Octavian and Valūnaitė Oleškevičienė, Giedrė and van Erp, Marieke, Ll(o)d and nlp perspectives on semantic change for humanities research, Semantic Web 13 (2022) 1051–1080. doi:10.3233/SW-222848.

[11] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs, in: M. Martin, M. Cuquet, E. Folmer (Eds.), Joint Proceedings of the Posters and Demos Track of the 12th Interna-

tional Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016, volume 1695 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1695/paper4.pdf.
[12] B. Schrader, What's the Difference Between an Ontology and a Knowledge Graph? (White Paper), Technical Report, Enterprise Knowledge, (consulted September 8, 2021). URL: https://enterprise-knowledge.com/whats-the-difference-between-an-ontology-and-a-knowledge-graph/.
[13] I. Robinson, J. Webber, E. Eifrem, Graph Databases, 2nd ed., O'Reilly Media, 2015.
[14] M. Rodriguez, P. Neubauer, Constructions from dots and lines, Bul. Am. Soc.Info. Sci. Tech. 36 (2010) 35–41.
[15] S. Ferilli, Integration strategy and tool between formal ontology and graph database technology, Electronics 10 (2021). URL: https://www.mdpi.com/2079-9292/10/21/2616. doi:10.3390/electronics10212616.
[16] K. Beelen, F. Nanni, M. Coll Ardanuy, K. Hosseini, G. Tolfo, B. McGillivray, When time makes sense: A historically-aware approach to targeted sense disambiguation, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 2751–2761. URL: https://aclanthology.org/2021.findings-acl.243. doi:10.18653/v1/2021.findings-acl.243.
[17] E. Manjavacas, L. Fonteyn, Adapting vs. Pre-training Language Models for Historical Languages, Journal of Data Mining and Digital Humanities NLP4DH (2022). URL: https://hal.inria.fr/hal-03592137. doi:10.46298/jdmdh.9152.
[18] O. Becher, L. Hollink, D. Elliott, Exploring concept representations for concept drift detection, in: A. Fensel, L. Daniele, L. Aroyo, V. de Boer, S. Darányi, O. Elloumi, R. García-Castro, L. Hollink, O. Inel, G. Kuys, M. Maleshkova, M. Merdan, A. Meroño-Peñuela, T. Moser, F. L. Keppmann, E. Kontopoulos, L. Petram, E. Scarrone, R. Verborgh (Eds.), Joint Proceedings of SEMANTiCS 2017 Workshops co-located with the 13th International Conference on Semantic Systems (SEMANTiCS 2017), Amsterdam, Netherlands, September 11 and 14, 2017, volume 2063 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017. URL: http://ceur-ws.org/Vol-2063/dal-paper4.pdf.
[19] A. Fokkens, S. ter Braake, E. Maks, D. Ceolin, On the semantics of concept drift: Towards formal definitions of semantic change, in: S. Darányi, L. Hollink, A. Meroño Peñuela, E. Kontopoulos (Eds.), Proceedings of Drift-a-LOD, 2016.
[20] M. Wevers, M. Koolen, Digital begriffsgeschichte: Tracing semantic change using word embeddings, Historical Methods: A Journal of Quantitative and Interdisciplinary History 53 (2020) 226–243.