# Glossary

**Knowledge Graph** A knowledge graph (KG) is a directed, labeled multi-relational graph where nodes typically represent either entities or the attributes of entities, and (labeled) edges represent either relationships between entity-entity pairs or properties of entities. The simplest way to serialize a KG is as a set of triples, where each triple is of the form $(h, r, t)$ and represents an edge in the graph.

**Ontology** Although open-world KGs like Wikidata and Freebase have existed for a while, domain-specific KGs often derive their semantics and constraints from an underlying ontology. Although a deep definition of an ontology is not within the scope of this book, a simple definition is that an ontology is a set of terms defining the domain of interest. In the KG community, this amounts to a graph-like structure that contains concepts and relationships. A special relationship, is-a, serves as the glue relationship between the KG and the ontology. Evaluating when one ontology is 'better' than another continues to be a hotly debated issue, since it is not clear how to measure the goodness of an ontology using purely objective metrics.

**Information Extraction** When constructing domain-specific KGs, information extraction (IE) is the first set of algorithms that must be applied. IE refers to a set of techniques for ingesting natural language or HTML (and also other heterogeneous data that are neither structured nor natural language) and extracting useful information from them, usually with reference to an underlying ontology. IE today is often broken up into at least three sub-problems, each of which is important enough and challenging enough in its own right: Named Entity Recognition, Relation Extraction and Event Extraction.

**Named Entity Recognition** Named Entity Recognition (NER) is the best known sub-problem in Information Extraction. Although 'anything' in principle could be a named entity, in practice, named entities constitute instances of ontological types like persons, locations, organizations, facilities etc. In domain-specific applications, named entities can be esoteric and highly dependent on the domain e.g., physical attributes may be more common in the e-commerce domain than in

M. Kejriwal, *Domain-Specific Knowledge Graph Construction*, SpringerBriefs
in Computer Science, https://doi.org/10.1007/978-3-030-12375-8

domains like politics, academia or medicine. Pre-trained NER systems are useful for extracting generic types, but IE techniques have to be applied to the domain-specific cases. Supervised, semi-supervised and unsupervised methods for IE currently exist, and more recently, deep learning and representation learning have become very popular for achieving state-of-the-art performance.

**Word Embedding** Feature engineering has always been a bottleneck in traditional machine learning pipelines, especially for natural language processing (NLP) applications like NER. In recent times however, word embedding models have emerged as an efficient and powerful means of vectorizing words, documents and even graphs into low-dimensional, continuous spaces. These vectors, when optimized using a relatively simple notion of context, yield remarkable insights in the vector space, such as analogies and semantic clustering. Multiple word embedding algorithms now exist, although the original innovations are still widely used. Word embeddings have generally been adopted in favor of heavily engineered feature pipelines across multiple application-oriented communities in machine learning and knowledge discovery.

**Relation Extraction** After NER, Relation Extraction (RE) is the next most important step in an IE pipeline, and is essential for knowledge graph construction. It is uncommon to extract n-ary relations with n greater than 2; even for binary relations, performance is relatively poor compared to NER. Relation Extraction can be framed as a classification problem assuming the entities have been correctly extracted. Other ways of framing the problem also exist. Similar to NER, deep learning has emerged as an important technique for tackling RE.

**Event Extraction** Event extraction is yet another IE sub-problem, but one that tends to be limited to certain domains and ontologies. Events are typically identified by ‘triggers’ e.g., the word ‘hit’ might trigger an ‘attack’ event type, and tend to involve multiple arguments and relations. It is not unreasonable to think of an event as a ‘second-order’ entity for that reason. It is generally believed that much more research is required on event extraction before performance will reach acceptable levels for broader consumption. Most event extraction papers still tend to focus on ontologies like ACE and CAMEO, which are broad but, by no means, complete. It is unknown whether any of the current techniques, including the state-of-the-art, would be able to adapt with relatively low overhead if a new (domain) ontology were to be introduced.

**Entity Resolution** Entity Resolution (ER) is the problem of algorithmically determining when two instances (‘entities’) in the KG are the same underlying entity. The problem has been around for more than 50 years, with patient linking and census being the earliest applications. ER has been studied under many guises, including record linkage, instance matching and deduplication. Just like many of the other techniques in this book, supervised, semi-supervised and unsupervised solutions exist. Performance of ER systems can vary widely depending both on the training regime, the amount of data available and the domain. On some domains, such as geopolitical events, ER continues to suffer from performance issues compared to more traditional domains like census and publications.

**Blocking** Blocking refers to an important class of algorithmic techniques that are almost always included as the first step in a typical two-step ER workflow. Blocking is defined (in the most general case) as inexpensive clustering of approximately similar entities. By doing such clustering in sub-quadratic time, exhaustive pairwise comparisons can be avoided, leading to significant savings even for moderately sized datasets. Blocking has been explored for many decades now, and more recently, the automatic learning of blocking keys has become an important topic of research. Blocking for KGs is not as well-studied as blocking for relational databases.

**Knowledge Graph Embedding** Similar to a word embedding, a knowledge graph embedding (KGE) can be used to embed entities and relationships in a KG into a low-dimensional continuous space. Most successful KGE models are translational models, such as TransE and TransD, and rely on the same kinds of analogical intuitions as traditional word embedding algorithms. Recently, a lot of research has been going into how to incorporate 'extra' information into the KGE models, be it external text corpora, ontologies, rules, temporal information etc. In general, there are empirical benefits to including more information into the KGE model, but one has to be careful about diluting the domain-specific value of the KG itself when bringing in generic external sources.

**Linked Data** Linked Data is a set of four principles that guide the publication of (structured) data on the Web. Linked Data has continued to become popular, leading to the Linking Open Data (LOD) project, which contains datasets published openly on the Web using Linked Data standards. LOD now includes hundreds of datasets, including DBpedia, which is derived from Wikipedia, as its central hub. It is also the backbone for the broader Semantic Web ecosystem. Several rich applications are powered by LOD.

**Knowledge Graph Ecosystem** In the broadest sense, a knowledge graph ecosystem is a community, rather than a collection of datasets. Like any community, such an ecosystem is guided by its own social norms and incentive structures. As the name suggests, a KG ecosystem is centered around using KGs as a prominent technology, but the definition of a KG, and even a domain, will differ slightly based on the ecosystem. Norms can be radically different, even for Web-based ecosystems. For example, schema.org, launched by search engines in the earlier part of this decade, encourages isolated knowledge fragments that can be embedded in HTML and easily found by search engines, while Linked Data emphasizes connectivity and is agnostic to search as a specific application. Lately, there have been efforts to map and possibly reconcile such ecosystems. Whether this is really possible will depend on both social and technological factors.