Information Extraction meets the Semantic Web: A Survey

Jose L. Martinez-Rodriguez [a], Aidan Hogan [b] and Ivan Lopez-Arevalo [a]
[a] *Cinvestav Tamaulipas, Ciudad Victoria, Mexico*
E-mail: {lmartinez,ilopez}@tamps.cinvestav.mx
[b] *Center for Semantic Web Research, Department of Computer Science, University of Chile, Chile*
E-mail: ahogan@dcc.uchile.cl

Abstract. We provide a comprehensive survey of the research literature that applies Information Extraction techniques in a Semantic Web setting. Works in the intersection of these two areas can be seen from two overlapping perspectives: using Semantic Web resources (languages/ontologies/knowledge-bases/tools) to improve Information Extraction, and/or using Information Extraction to populate the Semantic Web. In more detail, we focus on the *extraction* and *linking* of three elements: *entities*, *concepts* and *relations*. *Extraction* involves identifying (textual) mentions referring to such elements in a given unstructured or semi-structured input source. *Linking* involves associating each such mention with an appropriate disambiguated identifier referring to the same element in a Semantic Web knowledge-base (or ontology), in some cases creating a new identifier where necessary. With respect to *entities*, works involving (Named) Entity Recognition, Entity Disambiguation, Entity Linking, etc. in the context of the Semantic Web are considered. With respect to *concepts*, works involving Term Extraction, Keyword Extraction, Topic Modeling, Topic Labeling, etc., in the context of the Semantic Web are considered. Finally, with respect to *relations*, works involving Relation Extraction in the context of the Semantic Web are considered. The focus of the majority of the survey is on works applied to unstructured sources (text in natural language); however, we also provide an overview of works that develop custom techniques adapted for semi-structured inputs, namely markup documents and web tables.

Keywords: Information Extraction, Entity Linking, Keyword Extraction, Topic Modeling, Relation Extraction, Semantic Web

1. Introduction

The Semantic Web pursues a vision of the Web where increased availability of structured content enables higher levels of automation. Berners-Lee [18] described this goal as being to "*enrich human readable web data with machine readable annotations, allowing the Web's evolution as the biggest database in the world*". However, making annotations on information from the Web is a non-trivial task for human users, particularly if some formal agreement is required to ensure that annotations are consistent across sources. Likewise, there is simply too much information available on the Web – information that is constantly changing – for it to be feasible to apply manual annotation to even a significant subset of what might be of relevance.

While the amount of structured data available on the Web has grown significantly in the past years, there is still a significant gap between the coverage of structured and unstructured data available on the Web [234]. Mika referred to this as the *semantic gap* [193], whereby the demand for structured data on the Web outstrips its supply. For example, in analysis of the 2013 Common Crawl dataset, Meusel *et al.* [189] found that of the 2.2 billion webpages considered, 26.3% contained some structured metadata. Thus, despite initiatives like Linking Open Data [256], Schema.org [188,192] (promoted by Google, Microsoft, Yahoo, and Yandex) and the Open Graph Protocol [117] (promoted by Facebook), this "semantic gap" is still observable on the Web today [193,189].

As a result, methods to automatically extract or enhance the structure of various corpora have been a core topic in the context of the Semantic Web. Such processes are often based on Information Extraction methods, which in turn are rooted in techniques from areas such as Natural Language Processing, Machine Learning and Information Retrieval. The combination of techniques from the Semantic Web and from Information Extraction can be seen as two-way: on the one hand, Information Extraction techniques can be applied to populate the Semantic Web, while on the other hand, Semantic Web techniques can be applied to guide the Information Extraction process. In some cases, both aspects are considered together, where an existing Semantic Web ontology or knowledge-base is used to guide the extraction, which further populates the given ontology and/or knowledge-base (KB).[1]

In the past years, we have seen a wealth of research dedicated to Information Extraction in a Semantic Web setting. While many such papers come from within the Semantic Web community, many recent works have come from other communities, where, in particular, general-knowledge Semantic Web KBs – such as DBpedia [159], Freebase [25] and YAGO2 [128] – have been broadly adopted as references for enhancing Information Extraction tasks. Given the wide variety of works emerging in this particular intersection from various communities (sometimes under different nomenclatures), we see that a comprehensive survey is needed to draw together the techniques proposed in such works. Our goal is then to provide such a survey.

Survey Scope: This survey provides an overview of published works that directly involve both Information Extraction methods and Semantic Web technologies. Given that both are very broad areas, we must be rather explicit in our inclusion criteria.

With respect to Semantic Web technologies, to be included in the scope of a survey, a work must make non-trivial use of an ontology, knowledge-base, tool or language that is founded on one of the core Semantic Web standards: RDF/RDFS/OWL/SKOS/SPARQL.[2]

By Information Extraction method, we focus on the extraction and/or linking of three main elements from an (unstructured or semi-structured) input source.

1. *Entities:* anything with named identity, typically an individual (e.g., `Barack Obama`, `1961`).
2. *Concepts:* a conceptual grouping of elements. We consider two types of concepts:
   - *Classes*: a named set of individuals (e.g., `U.S. President(s)`);
   - *Topics*: categories to which individuals or documents relate (e.g, `U.S. Politics`).
3. *Relations:* an n-ary tuple of entities ($n \geq 2$) with a predicate term denoting the type of relation (e.g., `marry(Barack Obama,Michele Obama,Chicago)`.

More formally, we can consider entities as atomic elements from the domain, concepts as unary predicates[3], and relations as n-ary ($n \geq 2$) predicates. We take a rather liberal interpretation of *concepts* to include both classes based on set-theoretic subsumption of instances (e.g., OWL classes [125]), as well as topics that form categories over which broader/narrower relations can be defined (e.g., SKOS concepts [194]). This is rather a practical decision that will allow us to draw together a collective summary of works in the interrelated areas of Term Extraction, Keyword Extraction, Topic Modeling, etc., under one heading.

Returning to "*extracting* and/or *linking*", we consider the extraction process as identifying mentions referring to such entities/concepts/relations in the unstructured or semi-structured input, while we consider the linking process as associating a disambiguated identifier in a Semantic Web ontology/KB for a mention (possibly creating one if not already present and using it to disambiguate and link further mentions).

To summarize, this survey includes papers that:

- deal with extraction and/or linking of entities, concepts and/or relations,
- deal with some Semantic Web standard – namely RDF, RDFS or OWL – or a resource published or otherwise using those standards,
- have details published in a relevant workshop, conference or journal since 1999,
- consider extraction from either unstructured or semi-structured sources.

[1] Herein we adopt the convention that the term "ontology" refers primarily to *terminological knowledge*, meaning that it describes classes and properties of the domain, such as *person*, *knows*, *country*, etc. On the other hand, we use the term "KB" to refer to primarily "*assertional knowledge*", which describes specific entities (aka. individuals) of the domain, such as *Barack Obama*, *China*, etc.

[2] Works that simply mention general terms such as "semantic" or "ontology" may be excluded by this criteria if they do not also directly use or depend upon a Semantic Web standard.

[3] Concepts are thus nothing more than built-in relations for which dedicated extraction methods are applied.

For practical reasons, we further limit the scope to only include papers that we could find through our survey methodology, which will be discussed in detail later. We may include out-of-scope papers to the extent that they serve as important background for the in-scope papers: for example, it is important for an uninitiated reader to understand some of the core techniques considered in the traditional Information Extraction area and to understand some of the core standards and resources considered in the core Semantic Web area.

Information Extraction Tasks: The survey deals with various Information Extraction tasks. We now give an introductory summary of the main tasks considered (though we note that the survey will delve into each task in much more depth later):

Named Entity Recognition: demarcate the locations of mentions of entities in an input text:
- aka. *Entity Recognition*, *Entity Extraction*;
- e.g., in the sentence "<u>`Barack Obama`</u> `was born in` <u>`Hawaii`</u>", mark the underlined phrases as entity mentions.

Entity Linking: associate mentions of entities with an appropriate disambiguated KB identifier:
- involves, or is sometimes synonymous with, *Entity Disambiguation*;[4] often used for the purposes of *Semantic Annotation*;
- e.g., associate "<u>`Hawaii`</u>" with the DBpedia identifier `dbr:Hawaii` for the U.S. state (rather than the identifier for various songs or books by the same name).[5]

Term Extraction: extract the main phrases that denote concepts relevant to a given domain described by a text collection, sometimes inducing hierarchical relations between concepts;
- aka. *Terminology Extraction*, often used for the purposes of *Ontology Learning*;
- e.g., identify from a text on Oncology that "`breast cancer`" and "`melanoma`" are important concepts in the domain;
- optionally identify that both of the above concepts are specializations of "`cancer`";
- terms may be linked to a KB/ontology.

Keyphrase Extraction: extract the main phrases that categorize the subject/domain of a text (unlike term extraction, the focus is often on describing the document, not the domain);
- aka. *Keyword Extraction*, which is often generically applied to cover extraction of multi-word phrases; often used for the purposes of *Semantic Annotation*;
- e.g., identify that the keyphrases "`breast cancer`" and "`mammogram`" help to summarize the subject of a particular document;
- keyphrases may be linked to a KB/ontology.

Topic Modeling: Cluster words/phrases frequently co-occurring together in the same context; these clusters are then interpreted as being associated to abstract topics to which a text relates;
- aka. *Topic Extraction*;
- e.g., identify that words such as "`cancer`", "`breast`", "`doctor`", "`chemotherapy`" tend to co-occur frequently and thus indicate that a document containing many such occurrences is about a particular abstract topic.

Topic Labeling: For clusters of words identified as abstract topics, extract a single term or phrase that best characterizes the topic;
- aka. *Topic Identification*, esp. when linked with an ontology/KB identifier; often used for the purposes of *Text Classification*;
- e.g., identify that the topic { "`cancer`", "`breast`", "`doctor`", "`chemotherapy`" } is best characterized with the term "`cancer`" (potentially linked to `dbr:Cancer` for the disease and not, e.g., the astrological sign).

Relation Extraction: Extract potentially n-ary relations (for $n \geq 2$) from an unstructured (i.e., text) or semi-structured (e.g., HTML table) source;
- a goal of the area of *Open Information Extraction*;
- e.g., in the sentence "`Barack Obama was born in Hawaii`", extract the binary relation `wasBornIn(Barack Obama,Hawaii)`;
- binary relations may be represented as RDF triples after linking entities and linking the predicate to an appropriate property (e.g., mapping `wasBornIn` to the DBpedia property `dbo:birthPlace`);
- n-ary ($n \geq 3$) relations are often represented with a variant of *reification* [123,253].

Note that we will use a more simplified nomenclature {`Entity`,`Concept`,`Relation`} × {`Extraction`,`Linking`} as previously described to structure our survey with

[4] In some cases Entity Linking is considered to include both recognition and disambiguation; in other cases, it is considered synonymous with disambiguation applied after recognition.

[5] We use well-known IRI prefixes as consistent with the lookup service hosted at: `http://prefix.cc`. All URLs in this paper were last accessed on 2017/10/14.

the goal of grouping related works together; in particular works on Term Extraction, Keyphrase Extraction, Topic Modeling and Topic Labeling will be grouped under the heading of Concept Extraction and Linking.

Again we are only interested in such tasks in the context of the Semantic Web. Our focus is on unstructured (text) inputs, but we will also give an overview of methods for semi-structured inputs (markup documents and tables) towards the end of the survey.

Related Areas, Surveys and Novelty: There are a variety of areas that relate and overlap with the scope of this survey, and likewise there have been a number of previous surveys in these areas. We now discuss such areas and surveys, how they relate to the current contribution, and outline the novelty of the current survey.

As we will see throughout this survey, Information Extraction (IE) from unstructured sources – i.e., textual corpora expressed primarily in natural language – relies heavily on Natural Language Processing (NLP). A number of resources have been published within the intersection of NLP and the Semantic Web, where we can point, for example, to a recent book published by Maynard *et al.* [179] in 2016; note that we also provide a brief primer on the most important NLP techniques in a supplementary appendix, discussed later.

On the other hand, Data Mining involves extracting *patterns* inherent in a dataset. Example Data Mining tasks include, classification, clustering, rule mining, predictive analysis, outlier detection, recommendation, etc. Knowledge Discovery refers to a higher-level process to help users extract knowledge from raw data, where a typical pipeline involves selection of data, pre-processing and transformation of data, a Data Mining phase to extract patterns, and finally evaluation and visualization to aid users gain knowledge from the raw data and provide feedback. Some IE techniques may rely on extracting patterns from data, which can be seen as a Data Mining step[6]; however, Information Extraction need not use Data Mining techniques, and many Data Mining tasks – such as outlier detection – have only a tenuous relation to Information Extraction. A survey of approaches that combine Data Mining/Knowledge Discovery with the Semantic Web was published by Ristoski and Paulheim [246] in 2016.

With respect to our survey, both Natural Language Processing and Data Mining form part of the background of our scope, but as discussed, Information Extraction has a rather different focus to both areas, neither covering nor being covered by either.

On the other hand, relating more specifically to the intersection of Information Extraction and the Semantic Web, we can identify the following (sub-)areas:

Semantic Annotation: aims to annotate documents with entities, classes, topics or facts, typically based on an existing ontology/KB. Some works on Semantic Annotation fall within the scope of our survey as they include extraction and linking of entities and/or concepts (though not typically relations). A survey focused on Semantic Annotation was published by Uren *et al.* [280] in 2006.

Ontology-Based Information Extraction: refers to leveraging the formal knowledge of ontologies to guide a traditional Information Extraction process over unstructured corpora. Such works fall within the scope of this survey. A prior survey of Ontology-Based Information Extraction was published by Wimalasuriya and Dou [293] in 2010.

Ontology Learning: helps automate the (costly) process of ontology building by inducing an (initial) ontology from a corpus of domain-specific text. Ontology Learning also often includes *Ontology Population*, meaning that instance of concepts and relations are also extracted. Such works fall within our scope. A survey of Ontology Learning was provided by Wong *et al.* [295] in 2012.

Knowledge Extraction: aims to lift an unstructured (or semi-structured) corpus into an output described using a knowledge representation formalism (such as OWL). Thus Knowledge Extraction can be seen as Information Extraction but with a stronger focus on using knowledge representation techniques to model outputs. In 2013, Gangemi [101] provided an introduction and comparison of fourteen tools for Knowledge Extraction over unstructured corpora.

Other related terms such as “Semantic Information Extraction” [99], “Knowledge-Based Information Extraction” [129], “Knowledge-Graph Completion” [167], and so forth, have also appeared in the literature. However, many such titles are used specifically within a given community, whereas works in the intersection of IE and SW have appeared in many communities. For example, “Knowledge Extraction” is used predominantly by the SW community and not

[6] In fact, the title “Information Extraction” pre-dates that of the title “Data Mining” in its modern interpretation.

others.[7] Hence our survey can be seen as drawing together works in such (sub-)areas under a more general scope: works involving IE techniques in a SW setting.

Intended Audience: This survey is written for researchers and practitioners who are already quite familiar with the main SW standards and concepts – such as the RDF, RDFS, OWL and SPARQL standards, etc. – but are not necessarily familiar with IE techniques. Hence we will not introduce SW concepts (such as RDF, OWL, etc.) herein. Otherwise, our goal is to make the survey as accessible as possible. For example, in order to make the survey self-contained, in Appendix A we provide a detailed primer on some traditional NLP and IE processes; the techniques discussed in this appendix are, in general, not in the scope of the survey (since they do not involve SW resources) but are heavily used by works that fall in scope. We recommend readers unfamiliar with the IE area to read the appendix as a primer prior to proceeding to the main body of the survey. Knowledge of some core Information Retrieval concepts – such as TF–IDF, PageRank, cosine similarity, etc. – and some core Machine Learning concepts – such as logistic regression, SVM, neural networks, etc. – may be necessary to understand finer details, but not to understand the main concepts.

Nomenclature: The area of Information Extraction is associated with a diverse nomenclature that may vary in use and connotation from author to author. Such variations may at times be subtle and at other times be entirely incompatible. Part of this relates to the various areas in which Information Extraction has been applied and the variety of areas from which it draws influence. We will attempt to use generalized terminology and indicate when terminology varies.

Survey Methodology: For finding in-scope papers, our methodology begins with a definition of keywords appropriate to the section at hand. These keywords are broken into lists of IE-related (e.g., "`entity extraction`", "`entity linking`") and SW-related (e.g., "`ontology`", "`semantic`"), where we apply a conjunction of their products to create keyphrases for search (e.g., "`entity extraction ontology`"). Given the diverse terminology used in different communities, often we need to try many variants of keyphrases to capture as many papers as possible.

For each keyphrase, we perform a search on Google Scholar for related papers. Extracting lists of papers (numbering in the thousands in total), we initially apply a rough filter for relevance based on the title and type of publication; thereafter, we filter by abstract, and finally by the body of the paper's content.

This methodology may miss relevant papers and is considered as seeding an initial list of papers. We subsequently apply a more informal methodology to extend this set to cover further papers: while reading relevant papers, we take note of other works referenced in related works, works that cite more prominent relevant papers, and also check the bibliography of prominent authors in the area for other papers that they have written. Occasionally while reading papers, we discover important keywords that had not been anticipated at the start,[8] for which we perform further searches.

We provide further details of our survey online.[9] This webpage includes further details on our survey methodology, including: the search keywords used, the inclusion/exclusion criteria, the methodology used to filter initial search results and to find further relevant papers. The lists of papers found through these searches, along with their metadata and how they were labeled in terms of relevance, are also provided; furthermore, the webpage provides supplementary data used in some examples, as well as details of the venues and areas in which highlighted papers were published.

Survey Structure: The structure of the remainder of this survey is as follows:

Appendix A provides a primer on classical Information Extraction techniques for readers previously unfamiliar with the IE area.

Section 2 discusses extraction and linking of entities for unstructured sources.

Section 3 discusses extraction and linking of concepts for unstructured sources.

Section 4 discusses extraction and linking of relations for unstructured sources.

Section 5 discusses techniques adapted specifically for extracting entities/concepts/relations from semi-structured sources.

Section 6 concludes the survey with discussion.

[7] Here we mean "Knowledge Extraction" in an IE-related context. Other works on generating explanations from neural networks use the same term in an unrelated manner.

[8] As an example, while reading relation extraction papers, we realized that "`distant supervision`" was an important keyphrase and performed a separate search for such papers.

[9] `http://www.tamps.cinvestav.mx/~lmartinez/survey/`

2. Entity Extraction & Linking

Entity Extraction & Linking (EEL)[10] refers to identifying mentions of entities in a text and linking them to a reference KB provided as input.

Entity Extraction can be performed using an off-the-shelf Named Entity Recognition (NER) tool as used in traditional IE scenarios (see Appendix A.1); however such tools typically extract entities for limited numbers of types, such as persons, organizations, places, etc.; on the other hand, the reference KB may contain entities from hundreds of types. Hence, while some Entity Extraction & Linking tools rely on off-the-shelf NER tools, others define bespoke methods for identifying entity mentions in text, typically using entities labels in the KB as a dictionary to guide the extraction.

Once entity mentions are extracted from the text, the next phase involves linking – or *disambiguating* – these mentions by assigning them to KB identifiers; typically each mention in the text is associated with a single KB identifier chosen by the process as the most likely match, or is associated with multiple KB identifiers and an associated weight (aka. *support*) indicating confidence in the matches that allow the application to choose which entity links to trust.

Example: In Listing 1, we provide an excerpt of an EEL response given by the online DBpedia Spotlight demo[11] in JSON format. Within the result, the "`@URI`" attribute is the selected identifier obtained from DBpedia, the "`@support`" is a degree of confidence in the match, the "`@types`" list matching classes from the KB, the "`@surfaceForm`" represents the text of the entity mention, the "`@offset`" indicates the character position of the mention in the text, the "`@similarityScore`" indicates the strength of a match with the entity label in the KB, and the "`@percentageOfSecondRank`" indicates the ratio of the support computed for the first- and second ranked document thus giving an idea of the level of ambiguity.

Listing 1: DBpedia Spotlight EEL example

```
Input:  Bryan Cranston is an American actor.  He is
     ↪  known for portraying "Walter White" in the
     ↪ drama series Breaking Bad.

Output:
{
 "@text": "Bryan Cranston is an American actor.  He
      ↪  is known for portraying \"Walter White\"
      ↪ in the drama series Breaking Bad.",
 "@confidence": "0.35",
 "@support": "0",
 "@types": "",
 "@sparql": "",
 "@policy": "whitelist",
 "Resources":   [
  {
   "@URI": "http://dbpedia.org/resource/
        ↪ Bryan_Cranston",
   "@support": "199",
   "@types": "DBpedia:Agent,Schema:Person,Http://
        ↪ xmlns.com/foaf/0.1/Person,DBpedia:Person
        ↪ ",
   "@surfaceForm": "Bryan Cranston",
   "@offset": "0",
   "@similarityScore": "1.0",
   "@percentageOfSecondRank": "0.0"
  },
  {
   "@URI": "http://dbpedia.org/resource/
        ↪ United_States",
   "@support": "560750",
   "@types": "Schema:Place,DBpedia:Place,DBpedia:
        ↪ PopulatedPlace,Schema:Country,DBpedia:
        ↪ Country",
   "@surfaceForm": "American",
   "@offset": "21",
   "@similarityScore": "0.9940788480408",
   "@percentageOfSecondRank": "0.003612999020603"
  },
  {
   "@URI": "http://dbpedia.org/resource/Actor",
   "@support": "35596",
   "@types": "",
   "@surfaceForm": "actor",
   "@offset": "30",
   "@similarityScore": "0.9999710345342",
   "@percentageOfSecondRank": "2.433621943875E-5"
  },
  {
   "@URI": "http://dbpedia.org/resource/
        ↪ Walter_White_(Breaking_Bad)",
   "@support": "856",
   "@types": "DBpedia:Agent,Schema:Person,Http://
        ↪ xmlns.com/foaf/0.1/Person,DBpedia:Person,
        ↪ DBpedia:FictionalCharacter",
   "@surfaceForm": "Walter White",
   "@offset": "66",
   "@similarityScore": "0.9999999999753",
   "@percentageOfSecondRank": "2.471061675685E-11"
  },
  {
   "@URI": "http://dbpedia.org/resource/Drama",
   "@support": "6217",
   "@types": "",
   "@surfaceForm": "drama",
   "@offset": "87",
   "@similarityScore": "0.8446404328140",
   "@percentageOfSecondRank": "0.1565036704039"
  },
  {
   "@URI": "http://dbpedia.org/resource/
        ↪ Breaking_Bad",
   "@support": "638",
   "@types": "Schema:CreativeWork,DBpedia:Work,
        ↪ DBpedia:TelevisionShow",
   "@surfaceForm": "Breaking Bad",
```

[10] We note that naming conventions can vary widely: sometimes Named Entity Linking (NEL) is used; sometimes the acronym (N)ERD is used for (Named) Entity Recognition & Disambiguation; sometimes EEL is used as a synonym for NED; other phrases can also be used, such as Named Entity Extraction (NEE), or Named Entity Resolution, or variations on the idea of semantic annotation or semantic tagging (which we consider applications of EEL).

[11] http://dbpedia-spotlight.github.io/demo/

```
      "@offset": "100",
      "@similarityScore": "1.0",
      "@percentageOfSecondRank": "4.6189529850760E-23"
    }
  ]
}
```

Of course, the exact details of the output of an EEL process will vary from tool to tool, but such a tool will minimally return a KB identifier and the location of the entity mention; a support will also often be returned.

Applications: EEL is used in a variety of applications, such as *semantic annotation* [38], where entities mentioned in text can be further detailed with reference data from the KB; *semantic search* [275], where search over textual collections can be enhanced – for example, to disambiguate entities or to find categories of relevant entities – through the structure provided by the KB; *question answering* [279], where the input text is a user question and the EEL process can identify which entities in the KR the question refers to; detecting *emerging entities* [126], where entities that do not yet appear in the KB, but may be candidates for adding to the KB, are extracted.[12] EEL can also serve as the basis for later IE processes, such as topic modeling, relation extraction, etc., as will be discussed later.

Process: As stated by various authors [57,142,229, 232], The EEL process is typically composed of two main steps: *recognition*, where relevant entity mentions in the text are found; and *disambiguation*, where entity mentions are mapped to candidate identifiers with a final weighted confidence. Since these steps are (often) loosely coupled, this section surveys the various techniques proposed for the recognition task and thereafter disambiguation. First, however, we give an overview of highlighted EEL systems.

System overview: Before discussing EEL techniques in more detail, we introduce some of the main systems covered, including their overall purpose; the year of the most recent version described by a publication; and distinguishing features. We only include *highlighted* EEL systems that deal with a resource (e.g., a KB) using one of the Semantic Web standards; deal with EEL over plain text; have a publication offering system details; *and* are standalone systems.

We list each system in order of year, and thereafter alphabetically. We use system names where available; otherwise, we use author initials (indicated by italics). The highlighted systems are then as follows:

SemTag (2003) [74] performs EEL with respect to the TAP ontology for the purposes of semantically annotating hundreds of millions of webpages using custom disambiguation methods based on a custom KB-relatedness measure.

KIM (2004) [236] (*Knowledge and Information Management*) performs EEL with respect to the KIM Ontology for semantically annotating text documents using GATE and Lucene.

AIDA (2011) [129] applies EEL over YAGO2 using contextual features from Wikipedia and a novel collective disambiguation procedure based on constructing a similarity graph between candidate KB entities and their associated mentions.

DBpedia Spotlight (2011) [187] performs EEL with respect to DBpedia, performing disambiguation using contextual features from Wikipedia and keyword-based similarity measures.

SDA (2011) [39] (*Semantic Disambiguation Algorithm*) performs EEL with respect to DBpedia, also performing disambiguation using contextual features from Wikipedia and keyword-based similarity measures for English, Spanish and French.

KORE (2012) [127] (*Keyphrase Overlap Relatedness for Entity disambiguation*) extends the AIDA system with new disambiguation features (in particular, to use Locality Sensitive Hashing), performing EEL with respect to YAGO2 and Wikipedia.

LINDEN (2012) [260] (*Linking named entIties with kNowleDge basE via semaNtic knowledge*) assumes that entities have been extracted and focuses on the problem of disambiguation with respect to YAGO1; Wikipedia categories are used to determine relatedness of candidate entities.

NERSO (2012) [114] (*Named Entity Recognition using Semantic Open data*) performs EEL with respect to DBpedia, with contextual information from Wikipedia search, applying a graph-based disambiguation algorithm.

THD (2012) [76] (*Targeted Hypernym Discovery*) performs EEL with respect to DBpedia, where the rankings returned by Wikipedia Search API are used for the purposes of disambiguating entities.

NereL (2013) [263] (*NER+EL*) performs EEL with respect to Freebase using Wikipedia for contextual information where various features are used to construct a graph over which a collective disambiguation model is defined.

[12] Emerging entities are also sometimes known as Out-Of Knowledge-Base (OOKB) entities or Not In Lexicon (NIL) entities.

AGDISTIS (2014) [281] applies EEL with respect to any KB that provides appropriate entity labels (DBpedia and YAGO2 are used for testing); a graph-based disambiguation process using HITS is used to link entities to the given KB.

AIDA-Light (2014) [217] applies EEL with respect to YAGO2 and Wikipedia, focusing on scalability; a two-stage disambiguation process is proposed where initial EEL results are used to determine the topic of the text, within which further entities are disambiguated in a second step.

Babelfy (2014) [203] combines EEL and Word Sense Disambiguation (WSD) into a unified framework. The system combines Wikipedia, WordNet and BabelNet into a Semantic Network that serves as a multilingual reference KB.

ExPoSe (2014) [225] extends DBpedia Spotlight – particularly with the ability to detect emerging entities – performing EEL with respect to Freebase using contextual features from Wikipedia.

***GianniniCDS* (2015) [108]** disambiguate DBpedia entities with a strategy based on *Common Subsumers*: concepts that subsume pairs of RDF resources associated with entity mentions.

JERL (2015) [171] (*Joint Entity Recognition & Linking*) performs EEL with respect to the Freebase and Microsoft Satori KBs; extraction and linking tasks are modeled jointly for collective inference.

Kan-Dis (2015) [134] performs EEL (and WSD) with respect to DBpedia and Freebase using Wikipedia for contextual information, focusing in particular on the use of various graph-based similarity measures for joint disambiguation.

Weasel (2015) [278] applies EEL with respect to DBpedia and Wikipedia, combining several features such as TF–IDF, cosine similarity, PageRank, amongst others, to train an SVM classifier used subsequently for disambiguation.

ADEL (2016) [233] (*ADadptive Entity Linking*) applies EEL with respect to DBpedia and YAGO2 over news articles, using various contextual features – such as coreference and domain relevance – to disambiguate identifiers.

CohEEL (2016) [112] (*Coherent and Efficient named Entity Linking*) proposes a graph-based model for EEL with respect to YAGO, applying a random-walk strategy to disambiguate candidate entities.

DoSeR (2016) [312] (*Disambiguation of Semantic Resources*) focuses on disambiguating DBpedia entities, where variants of word embeddings are used to construct a similarity graph over which PageRank is applied to rank candidates.

J-NERD (2016) [218] (*Joint Named Entity Recognition and Disambiguation*) performs EEL with respect to YAGO2/WordNet and Wikipedia using a probabilistic graphical model that captures joint inference for extraction and disambiguation.

NERFGUN (2016) [115] performs collective entity disambiguation using low-level features to populate a probabilistic model – based on undirected factor graphs – over which inference is applied.

Table 1 provides an overview of the high-level EEL techniques used by each of these systems. These techniques will be surveyed in later sub-sections.

Excluded works: There are a number of popular EEL services and tools for which details cannot be found in the literature. Some of these are commercial systems that are closed source with undisclosed details. Others are open source projects that simply have not formally published their details. Such works are out-of-scope for this survey, but since many such systems are quite popular and relevant in practice for EEL-related tasks, we mention them here:

AlchemyAPI[13] is an IBM product that offers various NLP services, where EEL functionality is supported with respect to KBs such as DBpedia or FreeBase and output can be returned in RDF. The Alchemy API comes with a development kit that supports various programming languages and offers 1,000 daily operations to registered users.

Apache Stanbol[14] is an Open Source project that allows for extending existing Content Management Systems (CMSs) with semantic features, including methods for EEL (based on OpenNLP). The RESTful services it exposes provide support for RDF and JSON-LD formats.

OpenCalais[15] is a labeling tool for NER types such as people, place, companies, events, etc. It is composed of NLP taggers and machine learning algorithms. The resulting format output is RDF, JSON, or N3, and it allows up to 50,000 daily service requests per client.

TextRazor[16] provides a text analytics web service for spelling correction, entity extraction, topic tagging among other extraction tasks; its API allows 500 free daily requests.

[13] https://www.ibm.com/watson/alchemy-api.html

[14] https://stanbol.apache.org/

[15] http://www.opencalais.com/

[16] https://www.textrazor.com/

Table 1

Overview of Entity Extraction & Linking systems

KB denotes the main knowledge-base used; **Matching** and **Indexing** refer to methods used to match/index entity labels from the KB; **Context** refers to the sources of contextual information used; **Recognition** refers to the process for identifying entity mentions; **Disambiguation** refers to the types of high-level disambiguation features used (M:Mention, K:Keyword, G:Graph, C:Category, L:Linguistic); '—' denotes no information found, not used or not applicable

System	**Year**	**KB**	**Matching**	**Indexing**	**Context**	**Recognition**	**Disambiguation**
ADEL [233]	2016	DBpedia	Keyword	Elastic Couchbase	Wikipedia	Tokens Stanford POS Stanford NER	M,G
AGDISTIS [281]	2014	Any	Keyword	Lucene	Wikipedia	Tokens	M,G
AIDA [129]	2011	YAGO2	Keyword	Postgres	Wikipedia	Stanford NER	M,K,G
AIDA-Light [217]	2014	YAGO2	Keyword LSH	Dictionary LSH	Wikipedia	Tokens	M,K,G,C
Babelfy [203]	2014	Wikipedia WordNet BabelNet	Substring	—	Wikipedia	Stanford POS	M,G,L
CohEEL [112]	2016	YAGO2	Keywords	—	Wikipedia	Stanford NER	K, G
DBpedia Spotlight [187]	2011	DBpedia	Substring	Aho–Corasick	Wikipedia	LingPipe POS	M,K
DoSeR [312]	2016	DBpedia YAGO3	Exact Keywords	Custom	Wikipedia	—	M, G
ExPoSe [225]	2014	DBpedia	Substring	Aho–Corasick	Wikipedia	LingPipe POS	M,K
GianniniCDS [108]	2015	DBpedia	Substring	SPARQL	Wikipedia	—	C
JERL [171]	2015	Freebase Satori	—	—	Wikipedia	Hybrid (CRF)	K,G,C,L
J-NERD [218]	2016	YAGO2	Keyword	Dictionary	Wikipedia	Hybrid (CRF)	M,K,C,L
Kan-Dis [134]	2015	DBpedia Freebase	Keyword	Lucene	Wikipedia	Stanford NER	K,G,L
KIM [236]	2004	KIMO	Keyword	Hashmap	—	GATE JAPE	G
KORE [127]	2012	YAGO2	Keyword LSH	Postgres	Wikipedia	Stanford NER	M,K,G
LINDEN [260]	2012	YAGO1	—	—	Wikipedia	—	C
NereL [263]	2013	Freebase	Keyword	Freebase API	Wikipedia	UIUC NER Illinois Chunker Tokens	M,K,G,C,L
NERFGUN [115]	2016	DBpedia	Substring	Dictionary	Wikipedia	—	M, K, G
NERSO [114]	2012	DBpedia	Exact	SPARQL	Wikipedia	Tokens	G
SDA [39]	2011	DBpedia	Keyword	—	Wikipedia	Tokens	K
SemTag [74]	2003	TAP	Keyword	—	Lab. Data	Tokens	K
THD [76]	2012	DBpedia	Keyword	Lucene	Wikipedia	GATE JAPE	K,G
Weasel [278]	2015	DBpedia	Substring	Dictionary	Wikipedia	Stanford NER	K, G

PoolParty Semantic Suite[17] offers a variety of products, amongst which is a Text Mining & Entity Extraction tool used to support a variety of applications, such as semantic search, automatic content tagging, content recommendations, etc.

Yahoo! Content Analysis API[18] detects entities/concepts, categories, and relationships within unstructured content. It ranks those detected entities/concepts by their overall relevance, resolves those if possible into Wikipedia pages, and annotates tags with relevant meta-data.

Though referenced by various EEL papers, we could not find technical details on the Zemanta website[19] at the time of writing. Other commercial tools are available for NER, but to the best of our knowledge, do not support disambiguation and linking with respect to a KB.

We also exclude from the summary approaches that only consider Wikipedia as a KB (Wikiminer [196, 197], TAGME [91], Illinois Wikifier [240], Semanticizer [186], WAT [229], amongst others [79,174,104]) as a reference KB; however, given that linking to Wikipedia makes it trivial to link to DBpedia and other KBs, we will discuss such works in later subsections. We also exclude systems that rely on meta-data other than general text, for example, specializing in EEL over social networks or Tweets [299,300,72], video captions [191], spoken language [17], keyword search logs [58], etc.; some of these works will be discussed later in Section 5. We also exclude ensemble tools (e.g., Dexter [37], NERD [248], NTUNLP [44] and WESTLAB [38]) that combine results from multiple underlying tools; we will discuss these later. Finally we exclude a number of challenge papers that adapt existing tools to generate evaluation results rather than proposing novel techniques (e.g., [86,168,225]).

2.1. Extraction

The goal of EEL is to extract and link entity mentions in a text with entity identifiers in a KB; some tools may additionally detect and propose identifiers for emerging entities that are not yet found in the KB [225,233,218]. In both cases, the first step is to mark entity mentions in the text that can be linked (or proposed as an addition to) the KB. Thus traditional NER tools – discussed in Appendix A.1 – can be used. However, in the context of EEL where a target KB is given as input, there can be key differences between a typical EEL recognition phase and traditional NER.

- In cases where emerging entities are not detected, the KB can provide a full list of target entity labels, which can be stored in a *dictionary* that is used to find mentions of those entities. While dictionaries can be found in traditional NER scenarios, these often refer to individual tokens that strongly indicate an entity of a given type, such as common first or family names, lists of places and companies, etc. On the other hand, in EEL scenarios, the dictionary can be populated with complete entity labels from the KB for a wider range of types; in scenarios not involving emerging entities, this dictionary will be complete for the entities to recognize. Of course, this can lead to a very large dictionary, depending on the KB used.
- Relating to the previous point, (particularly) in scenarios where a complete dictionary is available, the line between extraction and linking can become blurred since labels in the dictionary from the KB will often be associated with KB identifiers; hence, dictionary-based detection of entities will also provide initial links to the KB. Such approaches are sometimes known as End-to-End (*E2E*) approaches [233], where extraction and linking phases become more tightly coupled.
- In traditional NER scenarios, extracted entity mentions are typically associated with a type, usually with respect to a number of trained types such as person, organization, and location. However, in many EEL scenarios, the types are already given by the KB and are in fact often much richer than what traditional NER models support.

In this section, we thus begin by discussing the preparation of a dictionary and methods used for recognizing entities in the context of EEL.

2.1.1. Dictionary

The predominant method for performing EEL relies on using a dictionary – also known as a *lexicon* or *gazetteer* – which maps labels of target entities in the KB to their identifiers; for example, a dictionary might map the label "`Bryan Cranston`" to the DBpedia IRI `dbr:Bryan_Cranston`. In fact, a single KB entity may have multiple labels (aka. *aliases*) that map to one identifier, such as "`Bryan Cranston`", "`Bryan Lee Cranston`", "`Bryan L. Cranston`", etc. Furthermore,

[17] `https://www.poolparty.biz/`

[18] `https://developer.yahoo.com/contentanalysis/`

[19] `http://www.zemanta.com/`

some labels may be ambiguous, where a single label may map to a set of identifiers; for example, "Boston" may map to dbr:Boston, dbr:Boston_(band), and so forth. Hence a dictionary may map KB labels to identifiers in a many-to-many fashion. Finally, for each KB identifier, a dictionary may contain contextual features to help disambiguate entities in a later stage; for example, context information may tell us that dbr:Boston is typed as dbo:City in the KB, or that known mentions of dbr:Boston in a text often have words like "population" or "metropolitan" nearby.

Thus, with respect to dictionaries, the first important aspect is the selection of entities to consider (or, indeed, the source from which to extract a selection of entities). The second important aspect – particularly given large dictionaries and/or large corpora of text – is the use of optimized indexes that allow for efficient matching of mentions with dictionary labels. The third aspect to consider is the enrichment of each entity in the dictionary with contextual information to improve precision of matches. We now discuss these three aspects of dictionaries in turn.

Selection of entities: In the context of EEL, an obvious source from which to form the dictionary is the labels of target entities in the KB. In many Information Extraction scenarios, KBs pertaining to general knowledge are employed; the most commonly used are:

DBpedia [159] A KB extracted from Wikipedia and used by ADEL [233], DBpedia Spotlight [187], ExPoSe [225], Kan-Dis [134], NERSO [114], Seznam [86], SDA [39] and THD [76], as well as works by Exner and Nugues [87], Nebhi [213], Giannini *et al.* [108], amongst others;

Freebase [25] A collaboratively-edited KB – previously hosted by Google but now discontinued in favour of Wikidata [274] – used by JERL [171], Kan-Dis [134], NEMO [63], Neofonie [146], NereL [263], Seznam [86], Tulip [168], as well as works by Zheng *et al.* [306], amongst others;

Wikidata [289] A collaboratively-edited KB hosted by the Wikimedia Foundation that, although released more recently than other KBs, has been used by HERD [266];

YAGO(2) [128] Another KB extracted from Wikipedia with richer meta-data, used by AIDA [129], AIDA-Light [217], CohELL [112], J-NERD [218], KORE [127] and LINDEN [260], as well as works by Abedini *et al.* [1], amongst others.

It is important to note that these KBs are tightly coupled with owl:sameAs links establishing KB-level coreference and are also tightly coupled with Wikipedia; this implies that once, for example, entities are linked to one such KB, they can be transitively linked to the other KBs mentioned, and vice versa.

Many of the entities in these KBs may be irrelevant for certain application scenarios. Some systems support selecting a subset of entities from the KB to form the dictionary, potentially pertaining to a given domain or a selection of types. For example, DBpedia Spotlight [187] can build a dictionary from the DBpedia entities returned as results for a given SPARQL query.

Other systems may use other specific KBs not mentioned, where for example, Babelfy [203] constructs its own KB from a unification of Wikipedia, WordNet, and BabelNet; JERL [171] uses a proprietary KB (Microsoft's Satori) alongside Freebase; SemTag [74] – which pre-dates all of the previously mentioned KBs – uses Stanford's TAP KB; KIM [236] creates a custom KB called KIMO; etc.

Dictionary matching and indexing: In order to match mentions with the dictionary in an efficient manner, optimized data structures are required, which depend on the form of matching employed. The need for efficiency is particularly important for some of the KBs previously mentioned, where the number of target entities involved can go into the millions. The size of the input corpora is also an important consideration: while slower (but potentially more accurate) matching algorithms can be tolerated for smaller inputs, such algorithms are impractical for larger input texts.

A major challenge is that desirable matches may not be an exact match, but may rather only be captured by an approximate string-matching algorithm. While one could consider, for example, approximate matching based on regular expressions or edit distances, such measures do not lend themselves naturally to index-based approaches. Instead, for large dictionaries, or large input corpora, it may be necessary to trade recall (i.e., the percentage of correct spots captured) for efficiency by using coarser matching methods. Likewise, it is important to note that KBs such as DBpedia enumerate multiple "alias" labels for entities (extracted from the redirect entries in Wikipedia), which if included in the dictionary, can help to improve recall while using coarser matching methods.

A popular approach to index the dictionary is to use some variation on a prefix tree (aka. trie), such as used by the Aho–Corasick string-searching algorithm, which can find mentions of an input list of strings within an input text in time linear to the combined size of the inputs and output. The main idea is to represent the dictionary as a prefix tree where nodes refer to letters, and transitions refer to sequences of letters in a dictionary word; further transitions are put from failed matches (dead-ends) to the node representing the longest matching prefix in the dictionary. With the dictionary preloaded into the index, the text can then be streamed through the index to find (prefix) matches. Typically phrases are indexed separately to allow both word-level and phrase-level matching. This algorithm is implemented by GATE and LingPipe, with the latter being used by DBpedia Spotlight [187].

The main drawback of tries is that, for the matching process to be performed efficiently, the dictionary index must fit in memory, which may be prohibitive for very large dictionaries. For these reasons, the Lucene/Solr Tagger implements a more general *finite state transducer* that also reuses suffixes and byte-encodings to reduce space [67]. Using this method, for example, the Lucene documentation claims that 9.8 million unique terms appearing in English Wikipedia can be indexed in 8 seconds using 256 MB of heap space, and is thus far more efficient than GATE's implementation[20]; this index is used by HERD [266] and Tulip [168] to store KB labels.

In other cases, rather than using traditional Information Extraction frameworks, some authors have proposed to implement custom indexing methods. To give some examples, KIM [236] uses a hash-based index over tokens in an entity mention[21]; AIDA-Light [217] uses a Locality Sensitive Hashing (LSH) index to find approximate matches in cases where an initial exact-match lookup fails; and so forth.

Of course, the problem of indexing the dictionary is closely related to the problem of inverted indexing in Information Retrieval, where keywords are indexed against the documents that contain them. Such inverted indexes have proven their scalability and efficiency in Web search engines such as Google, Bing, etc., and likewise support simple forms of approximate matching based on, for example, stemming or lemmatization, which pre-normalize document and query keywords. Exploiting this natural link to Information Retrieval, the ADEL [233], AGDISTIS [281], Kan-Dis [134], TAGME [91] and WAT [229] systems use inverted-indexing schemes such as Lucene[22] and Elastic[23].

To manage the structured data associated with entities, such as identifiers or contextual features, some tools use more relational-style data management systems. For example, AIDA [129] uses the PostgreSQL relational database to retrieve entity candidates, while ADEL [233] and Neofonie [146] use the Couchbase[24] and Redis [25] NoSQL stores, respectively, to manage the labels and meta-data of their dictionaries.

Contextual features: Rather than being a flat map of entity labels to (sets of) KB identifiers, dictionaries often include contextual features to later help disambiguate candidate links. Such contextual features may be categorized as being *structured* or *unstructured*.

Structured contextual features are those that can be extracted directly from a structured or semi-structured source. In the context of EEL, such features are often extracted from the reference KB itself. For example, each entity in the dictionary can be associated with the (labels of the) types of that entity, but also perhaps the labels of the properties that are defined for it, or a count of the number of triples associated with, or the entities it is related to, or its centrality (and thus "importance") in the graph-structure of the KB, and so forth.

On the other hand, unstructured contextual features are those that must instead be extracted from textual corpora. In most cases, this will involving extracting statistics and patterns from an external reference corpus that potentially has already had its entities labeled (and linked with the KB). Such features may capture patterns in text surrounding the mentions of an entity, entities that are frequently mentioned close together, patterns in the anchor-text of links to a page about that entity, in how many documents a particular entity is mentioned, how many times it tends to be mentioned in a particular document, and so forth; clearly such information will not be available from the KB itself.

A very common choice of text corpora for extracting both structured and unstructured contextual features is Wikipedia, whose use in this setting was – to the best of our knowledge – first proposed by Bunescu and Pasca [31], then later followed by many other subsequent works [62,91,39,240,36,37,229,232]. The

[20] http://blog.mikemccandless.com/2010/12/using-finite-state-transducers-in.html

[21] This implementation was later integrated into GATE: https://gate.ac.uk/sale/tao/splitch13.html

[22] http://lucene.apache.org/core/

[23] https://www.elastic.co; note that ElasticSearch is in fact based on Lucene

[24] http://www.couchbase.com

[25] https://redis.io/

widespread use of Wikipedia can be explained by the unique advantages it has for such tasks:

- The text in Wikipedia is primarily factual and available in a variety of languages.
- Wikipedia has broad coverage, with documents about entities in a variety of domains.
- Articles in Wikipedia can be directly linked to the entities they describe in various KBs, including DBpedia, Freebase, Wikidata, YAGO(2), etc.
- Mentions of entities in Wikipedia often provide a link to the article about that entity, thus providing labeled examples of entity mentions and associated examples of anchor text in various contexts.
- Aside from the usual textual features such as term frequencies and co-occurrences, a variety of richer features are available from Wikipedia that may not be available in other textual corpora, including disambiguation pages, redirections of aliases, category information, info-boxes, article edit history, and so forth.[26]

We will further discuss how such contextual features – stored as part of the dictionary – can be used for disambiguation later in this section.

2.1.2. *Recognition*

We now assume a dictionary that maps labels (e.g., "`Bryan Cranston`", "`Bryan Lee Cranston`", etc.) to a (set of) KB identifier(s) for the entity question (e.g,, "`dbr:Bryan_Cranston`") and potentially some contextual information (e.g., often co-occurs with "`dbr:Breaking_Bad`", anchor text often uses the term "`Heisenberg`", etc.). In the next step, we need to somehow identify entity mentions in the input text. We refer to this process as *recognition* – also known as *spotting* – where we now survey some key approaches.

Token-based: Given that entity mentions may consist of multiple sequential words – aka. *n*-grams – the brute-force option would be to send all *n*-grams in the input text to the dictionary, for *n* up to, say, the maximum number of words found in a dictionary entry, or a fixed parameter. We refer generically to these *n*-grams as *tokens* and to these methods for extracting *n*-grams as *tokenization*. Sometimes these methods are referred to as window-based spotting or recognition techniques.

A number of systems use such a form of tokenization. Dill *et al.* [74] proposed Semtag, which uses the TAP ontology for seeking entity mentions that match tokens from the input text. In AIDA-Light [217], AGDISTIS [281], Lupedia [191], and NERSO [114], the spotting process is carried out by taking sliding windows over the text for varying-length *n*-grams.

Although relatively straightforward, a fundamental weakness with such methods relates to performance: given a large text, the dictionary-lookup implementation will have to be very efficient to deal with the number of tokens a typical such process will generate, many of which will be irrelevant. While some basic features, such as capitalization, can also be taken into account to filter (some) tokens, still, not all mentions may have capitalization, and many irrelevant or incoherent entities can still be retrieved; for example, by decomposing the text "`New York City`", the second bi-gram may produce York City in England as a candidate, though (probably) irrelevant to the mention. Such entities are known as *overlapping entities*, where post-processing must be applied (discussed later).

POS-based: A natural way to try to improve upon lexical tokenization methods in End-to-End systems is to try use some initial understanding of the grammatical role of words in the text, where POS-tags are used in order to be more selective with respect to what tokens are sent to be matched against the dictionary.

A first idea is to use POS-tags to quickly filter individual words that are likely to be irrelevant, where traditional NLP/IE libraries can be used in a preprocessing step. For example, ADEL [233], AIDA [129], Babelfy [203] and WAT [229] use the Stanford POS-tagger to focus on extracting entity mentions from words tagged as NNP (proper noun, singular) and NNPS (proper noun, plural). DBpedia Spotlight [187], on the other hand, relies on LingPipe POS-tagging, where verbs, adjectives, adverbs, and prepositions from the input text are disregarded.

A fundamental weakness of such approaches is that certain entity mentions may involve individual words that are not nouns and may be disregarded by the system; this is particularly common for entity types not usually considered by traditional NER tools, including titles of creative works like "`Breaking Bad`".[27] Heuristics such as analysis of capitalization can be used in certain cases to prevent filtering useful words; however, in other cases where words are not capitalized, the process will likely fail to recognise such

[26] Information from info-boxes, disambiguation, redirects and categories are also represented in a structured format in DBpedia.

[27] See Listing 8 where "`Breaking`" is tagged VGB (verb gerund/-past participle) and "`Bad`" as JJ (adjective).

mentions unless further steps are taken. Along those lines, to improve recall, Babelfy [203] first uses a POS-tagger to identify nouns that match substrings of entity labels in the dictionary, and then checks the surrounding text of the noun to try to expand the entity mention captured (using a maximum window of five words).

Parser-based: Rather than developing custom methods, one could consider using more traditional NER techniques to identify entity mentions in the text. Such an approach could also be used, for example, to identify emerging entities not mentioned in the KB. However, while POS-tagging is generally quite efficient, applying a full constituency or dependency parse might be too expensive for large texts. On the other hand, recognizing entity mentions often does not require full parse trees (aka. deep parsing methods).

As a trade-off, in traditional NER, shallow-parsing methods are often applied: such methods annotate an initial grouping – or *chunking* – of individual words, materializing a shallow tier of the full parse-tree. In the context of NER, noun-phrase chunks (see Listing 9 for an example NP/noun phrase annotation) are particularly relevant. As such, many EEL systems use "traditional" NER tools to identify entity mentions.

Along these lines, the THD system [76] uses GATE's rule-based *Java Annotation Patterns Engine (JAPE)*, consisting of regular-expression–like patterns over sequences of POS tags; more specifically, to extract entity mentions, THD uses the JAPE pattern NNP+, which will capture sequences of one-or-more proper nouns. This can be used to approximately identify noun phrase chunks in a form of shallow parsing that incurs little overhead when compared to POS-tagging.

As discussed in Appendix A, machine learning methods have become increasingly popular in recent years for parsing and NER. Hoffert *et al.* [126] propose combining AIDA and YAGO2 with Stanford NER – using a pre-trained Conditional Random Fields (CRF) classifier – to identify emerging entities. Likewise, ADEL [233] and UDFS [71] also use Stanford NER, while JERL [171] uses a custom unified CRF model that simultaneously performs extraction and linking. On the other hand, WAT [229] relies on OpenNLP's NER tool based on a Maximum Entropy model. Going one step further, J-NERD [218] uses the dependency parse-tree (extracted using a Stanford parser), where dependencies between nouns are used to create a tree-based model for each sentence, which are then combined into a global model across sentences, which in turn is fed into a subsequent approximate inference process based on Gibbs sampling.

One limitation of using machine-learning techniques in this manner is that they must be trained on a specific corpus. While Stanford NER and OpenNLP provide a set of pre-trained models, these tend to only cover the traditional NER types of person, organization, location and perhaps one or two more (or a generic miscellaneous type). On the other hand, a KB such as DBpedia may contain thousands of entity types, where off-the-shelf models would only cover a fraction thereof. Custom models can, however, be trained using these frameworks given appropriately labeled data, where for example ADEL [233] additionally trains models to recognize professions, or where UDFS [71] trains for ten types on a Twitter dataset, etc. However, richer types require richly-typed labeled data to train on. One option is to use sub-class hierarchies to select higher-level types from the KB to train with [218]. Furthermore, as previously discussed, in EEL scenarios, the types of entities are often given by the KB and need not be given by the NER tool: hence, other "non-standard" types of entities can be labeled "`miscellaneous`" to (try) train for generic spotting.

On the other hand, a benefit of such machine-learning approaches is that they can significantly reduce the amount of lookups required on the dictionary since, unlike token-based methods, initial entity mentions can be detected independently of the KB dictionary. Likewise, such methods can be used to detect emerging entities not yet featured in the KB.

Hybrid: The techniques described previously are sometimes complementary, where a number of systems thus apply hybrid approaches combining various such techniques. One such system is ADEL [233], which uses a mix of three high-level recognition techniques: persons, organizations and locations are extracted using Stanford NER; mentions based on proper nouns are extracted using Stanford POS; and more challenging mentions not based on proper nouns are extracted using an (unspecified) dictionary approach; entity mentions produced by all three approaches are fed into a unified disambiguation and pruning phase. A similar approach is taken by the FOX (*Federated knOwledge eXtraction Framework*) [270], which uses ensemble learning to combine the results of four NER tools – namely Stanford NER, Illinois NET, Ottawa BalIE, and OpenNLP – where the resulting entity mentions are then passed through the AGDISTIS [281] tool to subsequently link them to DBpedia.

2.2. *Disambiguation*

We now assume that a list of candidates identifiers has been retrieved from the KB for each mention of interest using the techniques previously described. However, as previously discussed, some KB labels in the dictionary may be ambiguous and may refer to multiple candidate identifiers. Likewise, the mentions in the text may not exactly match any single label in the dictionary. In summary, an individual mention may be associated with multiple initial candidates from the KB, where a distinguishing feature of EEL systems is the disambiguation phase, where the goal is to decide which KB identifiers best match which mentions in the text. In order to achieve this, the disambiguation phase will typically involve various forms of *filtering* and *scoring* of the initial candidate identifiers, considering both the candidates for individual entity mentions, as well as (collectively) considering candidates proposed for entity mentions in a region of the text. Disambiguation may thus result in:

- mentions being pruned as irrelevant to the KB (or proposed as emerging entities),
- candidates being pruned as irrelevant to a mention, and/or
- candidates being assigned a score – called a *support* – for a particular mention.

In some systems, phases of pruning and scoring may interleave, while in others, scoring is first applied and then pruning is applied afterward.

A wide variety of approaches to disambiguation can be found in the EEL literature. Our goal, in this survey, is thus to organize and discuss the main approaches used thus far. Along these lines, we will first discuss some of the low-level features that can be used to help with the disambiguation process. Thereafter we discuss how these features can be combined to select a final set of mentions and candidates and/or to compute a support for each candidate on a mention.

2.2.1. *Features*

In order to organize the discussion of features used by EEL systems to perform disambiguation, we will divide features into five high-level categories:

Mention-based (M): Such features rely on information about the entity mention itself, such as its text, the type selected by an NER tool (where available), the confidence in the mention, the presence of overlapping mentions, or the presence of abbreviated mentions.

Keyword-based (K): Such features rely on collecting contextual keywords for candidates and/or mentions from reference sources of text (often using Wikipedia). Keyword-based similarity measures can then be applied over pairs or sets of contexts.

Graph-based (G): Such features rely on constructing a (weighted) graph representing mentions and/or candidates and then applying analyses over the graph, such as to determine cocitation measures, dense-subgraphs, distances, or centrality.

Category-based (C): Such features rely on categorical information that capture the high-level domain of mentions, candidates and/or the input text itself, where Wikipedia categories are often used.

Linguistic-based (L): Such features rely on the grammatical role of words, or on the grammatical relation between words or chunks in the text (as produced by traditional NLP tools).

These categories reflect the type of information from which the features are extracted and will be used to structure this section, allowing us to introduce increasingly more complex types of sources from which to compute features. However, we can also consider an orthogonal conceptualization of features based on what they say about mentions or candidates:

Mention-only (mo): A feature about the mention independent of other mentions or candidates.

Mention–mention (mm): A feature between two or more mentions independent of their candidates.

Candidate-only (co): A feature about a candidate independent of other candidates or mentions.

Mention–candidate (mc): A feature about the candidate of a mention independent of other mentions.

Candidate–candidate (cc): A feature between two or more candidates independent of their mentions.

Various (v): A feature that may involve multiple of the above, or map mentions and/or candidates to a higher-level (or latent) feature, such as domain.

In Table 2, we give a summary of the disambiguation features mentioned by the papers describing the highlighted EEL systems, where the enumerated features are annotated with the previous two categorizations. We will now discuss these features in more detail in order of the type of information they consider.

Mention-based: With respect to disambiguation, important initial information can be gleaned from the mentions themselves, both in terms of the text of the mention, the type selected by the NER tool (where

Table 2

Overview of disambiguation features used by EEL systems

(M:Metric-based, K:Keyword-based, G:Graph-based, C:Category-based, L:Linguistic-based)
(mo:mention-only, mm:mention–mention, mc:mention–candidate, co:candidate-only, cc:candidate–candidate; v:various)

System	**String similarity** – [M | mc]	**Type comparison** – [M | mc]	**Keyphraseness** – [M | mo]	**Overlapping mentions** – [M | mm]	**Abbreviations** – [M | mm]	**Mention–candidate contexts** – [K | mc]	**Candidate–candidate contexts** – [K | cc]	**Commonness / Prior** – [G | mc]	**Relatedness (Cocitation)** – [G | cc]	**Relatedness (KB)** – [G | cc]	**Centrality** – [G | co]	**Categories** – [C | v]	**Linguistic** – [L | v]
ADEL [233]	✓	✓		✓	✓	✓		✓			✓		
AGDISTIS [281]	✓			✓						✓	✓		
AIDA [129]			✓			✓		✓	✓	✓			
AIDA-Light [217]	✓					✓	✓					✓	
Babelfy [203]											✓		✓
CohEEL [112]						✓		✓		✓			
DBpedia Spotlight [187]	✓		✓			✓							
DoSeR [312]	✓							✓			✓		
ExPoSe [225]	✓	✓	✓			✓							
GianniniCDS [108]												✓	
JERL [171]			✓			✓	✓	✓	✓				✓
J-NERD [218]	✓	✓	✓		✓	✓	✓					✓	✓
Kan-Dis [134]						✓	✓			✓	✓	✓	✓
KIM [236]										✓			
KORE [127]			✓		✓	✓	✓	✓	✓	✓			
LINDEN [260]									✓			✓	
NereL [263]	✓		✓				✓	✓	✓	✓		✓	✓
NERFGUN [115]	✓		✓				✓			✓	✓		
NERSO [114]											✓		
SDA [39]						✓							
SemTag [74]						✓							
THD [76]						✓					✓		
Weasel [278]						✓			✓		✓		

available), and the related of the mention to other neighboring mentions in the text.

First, the text of mentions can be used for disambiguation. While recognition often relies on matching a mention to a dictionary, this process is typically implemented using various forms of indexes that allow for efficiently matching substrings (such as prefixes, suffixes or tokens) or full strings. However, once a smaller set of initial candidates has been identified, more fine-grained string-matching can be applied between the respective mention and candidate labels. For example, given a mention "`Bryan L. Cranston`" and two candidates with labels "`Bryan L. Reuss`" (as the longest prefix match) and "`Bryan Cranston`" (as a keyword match), one could apply an edit-distance measure to refine these candidates. Along these lines, for example, ADEL [233] and NERFGUN [115] use Levenshtein edit-distance, while DoSeR [312] and AIDA-Light [217] use a trigram-based Jaccard similarity.[28] A natural limitation of such a feature is that it will score different candidates with the same labels with precisely the same score; hence such features are typically combined with other disambiguation features.

Second, whenever the recognition phase produces a type for entity mentions independently of the types available in the KB – as typically happens when a traditional NER tool is used – this NER-based type can be compared with the type of each candidate in the KB. Given that relatively few types are produced by NER tools (without using the KB) – where the most widely accepted types are *person*, *organization* and *location* – these types can be mapped manually to classes in the KB, where standard class inference techniques can be applied to also capture candidates that are instances of more specific classes. We note that both ADEL [233] and J-NERD [218] incorporate such a feature (both recently proposed approaches). While this can be a useful feature for disambiguating some entities, the KB will often contain types not covered by the NER tool (at least using off-the-shelf pre-trained models).

Third, the recognition process itself may produce a score for a mention indicating a confidence that it is referring to a (named) entity; this can then be used as a feature in the disambiguation phase. A simple such feature may capture capitalization, where HERD [266] and Tulip [168] mark lower-case mentions as "tentative" in the disambiguation phase, indicating that they need stronger evidence during disambiguation not to be pruned. Another popular feature, called *keyphraseness* by Mihalcea and Csomai [190], measures the number or ratio of times the mention appears in the anchor text of a link in a contextual corpus such as Wikipedia; this feature is considered by AIDA [129], DBpedia Spotlight [187], NERFGUN [115], HERD [266], etc. As such, the confidence of a mention computed during recognition may become a feature for disambiguation.

Fourth, we already mentioned how spotting may result in overlapping entity mentions being recognized, where, for example, the mention "`York City`" may overlap with the mention "`New York City`". A natural approach to resolve such overlaps – used, for example, by ADEL [233], AGDISTIS [281], HERD [266], KORE [127] and Tulip [168] – is to try expand entity mentions to a maximal possible match. While this seems practical, some of the "nested" entity mentions may be worth keeping. Consider "`New York City Police Department`"; while this is a maximal (aka. *external*) entity mention referring to an organization, it may also be valuable to maintain the nested "`New York City`" mention. As such, in the traditional NER literature, Finkel and Manning [95] argued for Nested Named Entity Recognition, which preserves relevant overlapping entities. However, we are not aware of any work on EEL that directly considers relevant nested entities, though systems such as Babelfy [203] explicitly allow overlapping entities. In certain (probably rare) cases, ambiguous overlaps without a maximal entity may occur, such as for "`The third Man Ray exhibition`", where "`The Third Man`" may refer to a popular 1949 movie, whilst "`Man Ray`" may refer to an American artist; neither are nested nor external entities. Though there are works on NER that model such (again, probably rare) cases [170], we are not aware of EEL approaches that explicitly consider such cases.

Fifth, we can consider abbreviated forms of mentions where a "complete mention" is used to introduce an entity, which is thereafter referred to using a shorter mention. For example, a text may mention "`Jimmy Wales`" in the introduction, but in subsequent mentions, the same entity may be referred to as simply "`Wales`"; clearly, without considering the presence of the longer entity mention, the shorter mention could be erroneously linked to the country. In fact, this is a particular form of coreference, where short mentions, rather than pronouns, are used to refer to an entity in subsequent mentions. A number of approaches – such as those proposed by Cucerzan [62] or Dur-

[28] More specifically, each input string is decomposed into a set of 3-character substrings, where the Jaccard coefficient (the cardinality of the intersection over union) of both sets is computed.

rett and Klein [81] for linking to Wikipedia, as well as systems such as ADEL [233], AGDISTIS [281], KORE [127] and Seznam [86] linking to RDF KBs – try to map short mentions to longer mentions appearing earlier in the text. On the other hand, counterexamples appear to be quite common, where, for example, a text on Enzo Ferrari may simultaneously use "`Ferrari`" as a mention for the person and the car company he founded; automatically disambiguating individual mentions may then prove difficult in such cases. Hence, this feature will often be combined with context features, described in the following.

Keyword-based: A variety of keyword-based techniques from the area of Information Retrieval (IR) are relevant not only to the recognition process, but also to the disambiguation process. While recognition can be done efficiently at large scale using inverted indexes, for example, relevance measures can be used to help score and rank candidates. A natural idea is to consider a mention as a keyword query posed against a textual document created to describe each KB entity, where IR measures of relevance can be used to score candidates. A typical IR measure used to determine the relevance of a document to a given keyword query is *TF–IDF*, where the core intuition is to consider documents that contain more mentions (term-frequency: TF) of relatively rare keywords (inverse-document frequency: IDF) in the keyword query to be more relevant to that query. Another typical measure is to use *cosine similarity*, where documents (and keyword queries) are represented as vectors in a normalized numeric space (known as a Vector Space Model (VSM) that may use, for example, numeric TF–IDF values), where the similarity of two vectors can be computed by measuring the cosine of the angle between them.

Systems relying on such measures for disambiguation include: DBpedia Spotlight [187], which defines a variant called TF–ICF, where ICF denotes inverse-candidate frequency, considering the ratio of candidates that mention the term; THD [76], which uses the Lucene-based search API of Wikipedia implementing measures similar to TF–IDF; SDA [39], which builds a textual context for each KB entity from Wikipedia based on article titles, content, anchor text, etc., where candidates are ranked based on cosine-similarity; NERFGUN [115], which compares mentions against the Wikipedia abstracts referring to KB entities using cosine-similarity; and so forth.

A subtle variant on such approaches is to consider an extended textual context not only for the KB entities, but also for the mentions. For example, considering the input sentence "`Santiago frequently experiences strong earthquakes.`", although Santiago is an ambiguous label, the term "`earthquake`" will most frequently appear in connection with the Santiago de Chile, which can be determined by comparing the keywords in the context of the mention with keywords in the contexts of the candidates. Such a keyword-based approach is used by SemTag [74], which performs entity linking with respect to the TAP KB: however, rather than build a context from an external source such as Wikipedia, the system instead extracts a context from the text surrounding human-labeled instances of linked entity mentions in a reference text.

Other more modern approaches adopt a similar *distributional* approach – where words are considered similar by merit of appearing frequently in similar contexts – but using more modern techniques. Amongst these, CohEEL [112] build a statistical language model for each KB entity according to the frequency of terms appearing in its associated Wikipedia article, where the probability that the context of a mention will be generated if that mention refers to a particular entity KB is estimated based on the model and used for disambiguation. A related approach is used in the DoSeR system [312], where *word embeddings* are used for disambiguation: in such an approach, words are represented as vectors in a fixed-dimensional numeric space where words that often co-occur with similar words will have similar vectors, allowing, for example, to predict words according to their context; the DoSeR system then computes word embeddings for KB entities using known entity links to model the context in which those entities are mentioned in the text, which can subsequently be used to predict further mentions of such entities based on the mention's context.

Another related approach is to consider *collective assignment*: rather than disambiguating one mention at a time by considering mention–candidate similarity, the selection of a candidate for one mention can affect the scoring of candidates for another mention. For example, considering the sentence "`Santiago is the second largest city of Cuba`", even though Santiago de Chile has the highest *prior probability* to be the entity referred to by "`Santiago`" (being a larger city mentioned more often), one may find that Santiago de Cuba has the strongest relation to Cuba (mentioned nearby) than all other candidates for the mention "`Santiago`"—or, in other words, Santiago de Cuba is the most *coherent* with Cuba, which can override the higher prior probability of Santiago de Chile.

While this is similar to the aforementioned distributional approaches, a distinguishing feature of collective assignment is to consider not only surrounding keywords, but also candidates for surrounding entity mentions. A seminal such approach – for EEL with respect to Wikipedia – was proposed by Cucerzan [62], where a cosine-similarity measure is applied between not only the contexts of mentions and their associated candidates, but also between candidates for neighboring entity mentions; disambiguation then attempts to simultaneously maximize the similarity of mentions to candidates as well as the similarity amongst the candidates chosen for other nearby entity mentions.

This idea of collective assignment would become influential in later works linking entities to RDF-based KBs. For example, the KORE [127] system extended AIDA [129] with a measure called *keyphrase overlap relatedness*[29], where mentions and candidates are associated with a keyword context, and where the relatedness of two contexts is based on the Jaccard similarity of their sets of keywords; this measure is then used to perform a collective assignment. To avoid computing pair-wise similarity over potentially large sets of candidates, the authors propose to use locality-sensitive hashing, where the idea is to hash contexts into a space such that similar contexts will be hashed into the same region (aka. bucket), allowing relatedness to be computed for the mentions and candidates in each bucket. Collective assignment based on comparing the textual contexts of candidates would become popular in many subsequent systems, including AIDA-Light [217], JERL [171], J-NERD [218], KanDis [134], and so forth. Collective assignment is also the underlying principle underlying many of the graph-based techniques discussed in the following.

Graph-based: During disambiguation, useful information can be gained from the graph of connections between entities in a contextual source such as Wikipedia, or in the target KB itself. First, graphs can be used to determine the prior probability of a particular entity; for example, considering the sentence "`Santiago is named after St. James.`", the context does not directly help to disambiguate the entity, but applying links analysis, it could be determined that (with respect to a given reference corpus) the candidate entity most commonly spoken about using the mention "`Santiago`" is Santiago de Chile. Second, as per the previous example for Cucerzan's [62] keyword-based disambiguation – "`Santiago is the second largest city of Cuba`" – one may find that a collective assignment can override the higher prior probability of an isolated candidate to instead maintain a high relatedness – or *coherence* – of candidates selected in a particular part of text, where *similarity graphs* can be used to determine the coherence of candidates.

A variety of entity disambiguation approaches rely on the graph structure of Wikipedia, where a seminal approach was proposed by Medelyan *et al.* [185] and later refined by Milne and Witten [196]. The graph-structure of Wikipedia is used to perform disambiguation based on two main concepts: *commonness* and *relatedness*. Commonness is measured as the (prior) probability that a given entity mention is used in the anchor text to point to the Wikipedia article about a given candidate entity; as an example, one could consider that the plurality of anchor texts in Wikipedia containing the (ambiguous) mention "`Santiago`" would link to the article on Santiago de Chile; thus this entity has a higher commonness than other candidates. On the other hand, relatedness is a cocitation measure of coherence based on how many articles in Wikipedia link to the articles of both candidates: how many inlinking documents they share relative to their total inlinks. Thereafter, unambiguous candidates help to disambiguate ambiguous candidates for neighboring entity mentions based on relatedness, which is weighted against commonness to compute a support for all candidates; the authors argue that the relative balance between relatedness and commonness depends on the context, where for example if "`Cuba`" is mentioned close to "`Santiago`", and "`Cuba`" itself has high commonness and low ambiguity, then this should override the commonness of Santiago de Chile since the context clearly relates to Cuba not Chile.

Further approaches then built upon and refined Milne and Witten's notion of *commonness* and *relatedness*. For example, Kulkarni *et al.* [156] propose a collective assignment method based on two types of score: a compatibility score defined between a mention and a candidate, computed using a selection of standard keyword-based approaches; and Milne and Witten's notion of *relatedness* defined between pairs of candidates. The goal then is to find the selection of candidates (one per mention) that maximizes the sum of the compatibility scores and all pairwise relatedness scores amongst selected candidates. While this optimization problem is NP-hard, the authors propose to use approximations based on integer linear programming and hill climbing algorithms.

[29] ... not to be confused with overlapping mentions.

Another approach using the notions of *commonness* and *relatedness* is that of TAGME [91]; however, rather than relying on the relatedness of unambiguous entities to disambiguate a context, TAGME instead proposes a more complex voting scheme, where the candidates for each entity can vote for the candidates on surrounding entities based on relatedness; candidates with higher commonness have stronger votes. Candidates with a commonness below a fixed threshold are pruned where two algorithms are then used to decide final candidates: Disambiguation by Classifier (DC), which uses commonness and relatedness as features to classify correct candidates; and Disambiguation by Threshold (DT), which selects the top-ε candidates by relatedness and then chooses the remaining candidate with the highest commonness (experimentally, the authors deem $\varepsilon = 0.3$ to offer the best results).

While the aforementioned tools link entity mentions to Wikipedia, other approaches linking to RDF-based KBs have followed adaptations of such ideas. One such tool is AIDA [129], which performs two main steps: *collective mapping* and *graph reduction*. In the collective mapping step, the tool creates a weighted graph that includes mentions and initial candidates as nodes: first, mentions are connected to their candidates by a weighted edge denoting their similarity as determined from a keyword-based disambiguation approach; second, entity candidates are connected by a weighted edge denoting their relatedness based on (1) the same notion of relatedness introduced by Milne and Witten [196], combined with (2) the distance between the two entities in the YAGO KB. The resulting graph is referred to as the *mention–entity graph*, whose edges are weighted in a similar manner to the measures considered by Kulkarni *et al.* [156]. In the subsequent graph reduction phase, the candidate nodes with the lowest weighted degree in this graph are pruned iteratively while preserving at least one candidate entity for each mention, resulting in an approximation of the densest possible (disambiguated) subgraph.

The general concept of computing a dense subgraph of the mention–entity graph was later reused in other systems. For example, the AIDA-Light [217] system (a variant of AIDA with focus on efficiency) uses keyword-based features to determine the weights on mention–entity and entity–entity edges in the mention–entity graph, from which a subgraph is then computed. As another variant on the dense subgraph idea, Babelfy [203] constructs a mention–entity graph but where edges between entity candidates are assigned based on semantic signatures computed using Random Walk with Restart over a weighted version of a custom semantic network (BabelNet); thereafter, an approximation of the densest subgraph is extracted by iteratively removing the least coherent vertices – considering the fraction of mentions connected to a candidate and its degree – until the number of candidates for every mention is below a specified threshold.

Rather than trying to compute a dense subgraph of the mention–entity graph, other approaches instead use standard centrality measures to score nodes in various forms of graph induced by the candidate entities. NERSO [114] constructs a directed graph of entity candidates retrieved from DBpedia based on the links between their articles on Wikipedia; over this graph, the system applies a variant on a *closeness centrality measure*, which, for a given node, is defined as the inverse of the average length of the shortest path to all other reachable nodes; for each mention, the centrality, degree and type of node is then combined into a final support for each candidate. On the other hand, the WAT system [229] extends TAGME [91] with various features, including a score based on the PageRank[30] of nodes in the mention–entity graph, which loosely acts as a context-specific version of the commonness feature. ADEL [233] likewise considers a feature based on the PageRank of entities in the DBpedia KB, while HERD [266], DoSeR [312] and Seznam [86] use PageRank over variants of a mention–entity graph. Using another popular centrality-based measure, AGDISTIS [281] first creates a graph by expanding the neighborhood of the nodes corresponding to candidates in the KB up to a fixed width; the approach then applies Kleinberg's HITS algorithm, using the authority score to select disambiguated entities.

In a variant of the centrality theme, Kan-Dis [134] uses two graph-based measures. The first measure is a baseline variant of Katz's centrality applied over the candidates in the KB's graph [224], where a parametrized sum over the k shortest paths between two nodes is taken as a measure of their relatedness such that two nodes are more similar the shorter the k shortest paths between them are. The second measure is then a weighted version of the baseline, where edges on paths are weighted based on the number of similar edges from each node, such that, for example, a path between two nodes through a country for the relation "resident" will have less effect on the overall re-

[30] PageRank is itself a variant of *eigenvector centrality*, which can be conceptualized as the probability of being at a node after an arbitrarily long random walk starting from a random node.

latedness of those nodes than a more "exclusive" path through a music-band with the relation "member".

Other systems apply variations on this theme of graph-based disambiguation. KIM [236] selects the candidate related to the most previously-selected candidates by some relation in the KB; DoSeR [312] likewise considers entities as related if they are directly connected in the KB and considers the degree of nodes in the KB as a measure of commonness; and so forth.

Category-based: Rather than trying to measure the coherence of pairs of candidates through keyword contexts or cocitations or their distance in the KB, some works propose to map candidates to higher-level category information and use such categories to determine the coherence of candidates. Most often, the category information of Wikipedia is used.

The earliest approaches to use such category information were those linking mentions to Wikipedia identifiers. For example, in cases where the keyword-based contexts of candidates contained insufficient information to derive reliable similarity measures, Bunescu and Pasca [31] propose to additionally use terms from the article categories to extend these contexts and learn correlations between keywords appearing in the mention context and categories found in the candidate context. A similar such idea – using category information from Wikipedia to enrich the contexts of candidates – was also used by Cucerzan [62].

A number of approaches follow similar intuitions to link entities to RDF KBs. One of the earliest such proposals was the LINDEN [260] approach, which was based on constructing a graph containing nodes representing candidates in the KB, their contexts, and their categories; edges are then added connecting candidates to their contexts and categories, while categories are connected by their taxonomic relations. Contextual and categorical information was taken from Wikipedia. A cocitation-based notion of candidate–candidate relatedness similar to that of Medelyan *et al.* [185] is combined with another candidate–candidate relatedness measure based on the probability of an entity in the KB falling under the most-specific shared ancestor of the categories of both entities.

As previously discussed, AIDA-Light [217] determines mention–candidate and candidate–candidate similarities using a keyword-based approach, where the similarities are used to construct a weighted mention–entity graph; this graph is also enhanced with categorical information from YAGO (itself derived from Wikipedia and WordNet), where category nodes are added to the graph and connected to the candidates in those categories; additionally, weighted edges between candidates can be computed based on their distance in the categorical hierarchy. J-NERD [218] likewise uses similar features based on latent topics computed from Wikipedia's categories.

Linguistic-based: Some more recent approaches propose to apply *joint inference* to combine disambiguation with other forms of linguistic analysis. Conceptually the idea is similar to that of using keyword contexts, but with a deeper analysis that also considers further linguistic information about the terms forming the context of a mention or a candidate.

We have already seen examples of how the recognition task can sometimes gain useful information from the disambiguation task. For example, in the sentence "`Nurse Ratched is a character in Ken Kesey's novel One Flew over the Cuckoo's Nest`", the latter mention – "`One Flew over the Cuckoo's Nest`" – is a challenging example for recognition due to its length, broken capitalization, uses of non-noun terms, and so forth; however, once disambiguated, the related entities could help to find recognize the right boundaries for the mention. As another example, in the sentence "`Bill de Blasio is the mayor of New York City`", disambiguating the latter entity may help recognize the former and vice versa (e.g., avoiding demarcating "`Bill`" or "`York City`" as mentions).

Recognizing this interdependence of recognition and disambiguation, one of the first approaches proposed to perform these tasks jointly was NereL [263], which applies a first high-recall NER pass that both underestimates and overestimates (potentially overlapping) mention boundaries, where features of these candidate mentions are combined with features for the candidate identifiers for the purposes of a joint inference step. A more complex unified model was later proposed by Durrett [81], which captured features not only for recognition (POS-tags, capitalization, etc.) and disambiguation (string-matching, PageRank, etc.), but also for coreference (type of mention, mention length, context, etc.), over which joint inference is applied. JERL [171] also uses a unified model for representing the NER and NED tasks, where word-level features (such as POS tags, dictionary hits, etc.) are combined with disambiguation features (such as commonness, coherence, categories, etc.), subsequently allowing for joint inference over both. J-NERD [218] likewise uses features based on Stanford's POS tagger and dependency parser, dictionary hits, coherence, categories, etc., to represent recognition and disambiguation in a unified model for joint inference.

Aside from joint recognition and disambiguation, other types of unified models have also been proposed. Babelfy [203] applies a joint approach to model and perform Named Entity Disambiguation and Word Sense Disambiguation in a unified manner. As an example, in the sentence "`Boston is a rock group`", the word "`rock`" can have various senses, where knowing that in this context it is used in the sense of a music genre will help disambiguate "`Boston`" as referring to the music group and not the city; on the other hand, disambiguating the entity "`Boston`" can help disambiguate the word sense of "`rock`", and thus we have an interdependence between the two tasks. Babelfy thus combines candidates for both word senses and entity mentions into a single *semantic interpretation graph*, from which (as previously mentioned) a dense (and thus coherent) sub-graph is extracted. Another approach applying joint Named Entity Disambiguation and Word Sense Disambiguation is Kan-Dis [134], where nouns in the text are extracted and their senses modeled as a graph – weighted by the notion of semantic relatedness described previously – from which a dense subgraph is extracted.

Summary of features: Given the breadth of features covered, we provide a short recap of the main features for reference:

Mention-based: Given the initial set of mentions identified and the labels of their corresponding candidates, we can consider:

- A mention-only feature produced by the NER tool to indicate the confidence in a particular mention;
- Mention–candidate features based on the string similarity between mention and candidate labels, or matches between mention (NER) and candidate (KB) types;
- Mention–mention features based on overlapping mentions, or the use of abbreviated references from a previous mention.

Keyword-based: Considering various types of textual contexts extracted for mentions (e.g., varying length windows of keywords surrounding the mention) and candidates (e.g., Wikipedia anchor texts, article texts, etc.), we can compute:

- Mention–candidate features considering various keyword-based similarity measures over their contexts (e.g., TF–IDF with cosine similarity; Jaccard similarity, word embeddings, and so forth);
- Candidate–candidate features based on the same types of similarity measures over candidate contexts.

Graph-based: Considering the graph-structure of a reference source such as Wikipedia, or the target KB, we can consider:

- Candidate-only features, such as prior probability based on centrality, etc.;
- Mention–candidate features, based on how many links use the mention's text to link to a document about the candidate;
- Candidate–candidate coherence features, such as cocitation, distance, density of subgraphs, topical coherence, etc.

Category-based: Considering the graph-structure of a reference source such as Wikipedia, or the target KB, we can consider:

- Candidate–category features based on membership of the candidate to the category;
- Text–category coherence features based on categories of candidates;
- Candidate–candidate features based on taxonomic similarity of associated categories.

Linguistic-based: Considering POS tags, word senses, coreferences, parse trees of the input text, etc., we can consider:

- Mention-only features based on POS or other NER features;
- Mention–mention features based on dependency analysis, or the coherence of candidates associated with them;
- Mention–candidate features based on coherence of sense-aware contexts;
- Candidate–candidate features based on connection through semantic networks.

This list of useful features for disambiguation is by no means complete and has continuously expanded as further entity linking papers have been published. Furthermore, EEL systems may use features not covered, typically exploiting specific information available in a particular KB, a particular reference source, or a particular input source. As some brief examples, we can mention that NEMO [63] uses geo-coordinate information extracted from Freebase to determine a geographical coherence over candidates, Yerva *et al.* [299] consider features computed from user profiles on Twitter and other social networks, ZenCrowd [70] considers features drawn from crowdsourcing, etc.

2.2.2. Scoring and pruning

As we have seen, a wide range of features have been proposed for the purposes of the disambiguation task. A general question then is: how can such features be weighted and combined into a final selection of candidates, or a final support for each candidate?

The most straightforward option is to consider a high-level feature used to score candidates (potentially using other features on a lower level), where for example AGDISTIS [281] relies on final HITS authority scores, DBpedia Spotlight [187] on TF–ICF scores, NERSO [114] on closeness centrality and degree; THD [76] on Wikipedia search rankings, etc.

Another option is to parameterize weights or thresholds for features and find the best values for them individually over a labeled dataset, which is used, for example, by Babelfy [203] to tune the parameters of its Random-Walk-with-Restart algorithm and the number of candidates to be pruned by its densest-subgraph approximation, or by AIDA [129] to configure thresholds and weights for prior probabilities and coherence.

An alternative method is to allow users to configure such parameters themselves, where AIDA [129] and DBpedia Spotlight [187] offer users the ability to configure parameters and thresholds for prior probabilities, coherence measures, tolerable level of ambiguity, and so forth. In this manner, a human expert can configure the system for a particular application, for example, tuning to trade precision for recall, or vice-versa.

Yet another option is to define a general objective function that then turns the problem of selecting the best candidates into an optimization problem, allowing the final candidate assignment to be (approximately) inferred. One such method is Kulkarni *et al.*'s [156] collective assignment approach, which uses integer linear programming and hill-climbing methods to compute a candidate assignment that (approximately) maximizes mention–candidate and candidate–candidate similarity weights. Another such method is JERL [171], which models entity recognition and disambiguation in a joint model over which dynamic programming methods are applied to infer final candidates. Systems optimizing for dense entity–mention subgraphs – such as AIDA [129], Babelfy [203] or KanDis [133] – follow similar techniques.

A final approach is to use classifiers to learn appropriate weights and parameters for different features based on labeled data. Amongst such approaches, we can mention that ADEL [233] performs experiments with *k*-NN, Random Forest, Naive Bayes and SVM classifiers, finding *k*-NN to perform best; AIDA [129], LINDEN [260] and WAT [229] use SVM variants to learn feature weights; HERD [266] uses logistic regression to assign weights to features; and so forth. All such methods rely on labeled data to train the classifiers over; we will discuss such datasets later when discussing the evaluation of EEL systems.

Such methods can be used to compute a final set of mentions and their candidates, either selecting a single candidate for each mention, or associating multiple with a support by which they can be ranked.

2.3. Ensemble systems

Entity linking tools can also be combined in an ensemble approach (as seen elsewhere, for example, in Machine Learning algorithms [73]). Different EEL systems may be suited to particular domains of text, or particular types of KBs, or may make varying tradeoffs between precision and recall, or may provide different types of auxiliary information as part of the EEL process, and so forth. The main goal of ensemble methods is to thus try to compare and exploit complementary aspects of the underlying systems such that the results obtained are better than possible using any single such system. Five such ensemble systems are:

NERD (2012) [248] (*Named Entity Recognition and Disambiguation*) uses an ontology to integrate the input and output of ten NER and EEL tools, namely AlchemyAPI, DBpedia Spotlight, Evri, Extractiv, Lupedia, OpenCalais, Saplo, Wikimeta, Yahoo! Content Extractor, and Zemanta. Later works proposed classifier-based methods (Naive Bayes, *k*-NN, SVM) for combining results [249].

BEL (2014) [310] (*Bagging for Entity Linking*) Recognizes entity mentions through Stanford NER, later retrieving entity candidates from YAGO that are disambiguated by means of a majority-voting algorithm according to various ranking classifiers applied over the mentions' contexts.

Dexter (2014) [37] uses TAGME and Wikiminer combined with a collective linking approach to match entity mentions in a text with Wikipedia identifiers, where they propose to be able to switch approaches depending on the features of the input document(s), such as domain, length, etc.

NTUNLP (2014) [44] performs EEL with respect to Freebase using a combination of DBpedia Spotlight and TAGME results, extended with a custom EEL method using the Freebase search API. Thresholds are applied over all three methods and overlapping mentions are filtered.

WESTLAB (2016) [38] uses Stanford NER & ADEL to recognize entity mentions, subsequently merging the output of four linking systems, namely AIDA, Babelfy, DBpedia Spotlight and TAGME.

2.4. Evaluation

EEL involves two high-level tasks: recognition and disambiguation. Thus evaluation may consider the recognition phase separately, or the disambiguation phase separately, or the entire EEL process as a whole. Given that the evaluation of recognition is well-covered by the traditional NER literature, here we focus on evaluations that consider whether or not the recognized mentions are deemed correct *and* whether or not the assigned KB identifier is deemed correct.

Given the wide range of EEL approaches proposed in the literature, we do not discuss details of the evaluations of individual tools conducted by the authors themselves. Rather we will discuss some of the most commonly used evaluation datasets and then discuss evaluations conducted by third parties to compare various selections of EEL systems.

2.4.1. Datasets

A variety of datasets have been used to evaluate the EEL process in different settings and under different assumptions. Here we enumerate some of the datasets that have been used to evaluate multiple tools:

AIDA–CoNLL [129]: The CoNLL-2013 dataset[31] consists of 1,393 Reuters' news articles whose entities were manually identified and typed for the purposes of training and evaluating traditional NER tools. For the purposes of training and evaluating AIDA [129], the authors manually linked the entities to YAGO. This dataset was later used by ADEL [233], AIDA-Light [217], Babelfy [203], HERD [266], JERL [171], J-NERD [218], KORE [127], amongst others.

AQUAINT [196] The AQUAINT dataset contains 50 English documents collected from the Xinhua News Service, New York Times, and the Associated Press. Each document contains about 250–300 words, where the first mention of an entity is manually linked to Wikipedia. The dataset was first proposed and used by Milne and Witten [196], and later used by AGDISTIS [281].

IITB [156] The IITB dataset contains 103 English webpages taken from a handful of domains relating to sports, entertainment, science and technology; the text of the webpages is scraped and semi-automatically linked with Wikipedia. The dataset was first proposed and used by Kulkarni [156] and later used by AGDISTIS [281].

Meij [186] This dataset contains 562 manually annotated tweets sampled from 20 "verified users" on Twitter and linked with Wikipedia. The dataset was first proposed by Meij *et al.* [186], and later used by Cornolti *et al.* [57] to form part of a more general purpose EEL benchmark.

MSNBC [62] The MSNBC dataset contains 20 English news articles from 10 different categories, which were semi-automatically annotated. The dataset was proposed and used by Cucerzan [62], and later reused to evaluate AGDISTIS [281], LINDEN [260] and by Kulkarni *et al.* [156].

KORE-50 [127] The KORE-50 dataset contains 50 English sentences designed to offer a challenging set of examples for entity linking tools; the sentences relate to various domains, including celebrities, music, business, sports, and politics. The dataset emphasizes short sentences, highly dense entity mentions, highly ambiguous mentions, and entities with low prior probability. The dataset was first proposed and used for KORE [127], and later reused by Babelfy [203], Kan-Dis [134], amongst others.

MEANTIME [199] The MEANTIME dataset consists of 120 English Wikinews articles on topics relating to finance, with translations to Spanish, Italian and Dutch. Entities are annotated with links to DBpedia resources. This dataset has been recently used by ADEL [233].

WP [127] The WP dataset samples English Wikipedia articles relating to heavy metal musical groups. Articles with related categories are retrieved and sentences with at least three named entities (found by anchor text in links) are kept; in total, 2019 sentences are considered. The dataset was first proposed and used for the KORE [127] system and also later used by AIDA-Light [217].

Aside from being used for evaluation, we note that such datasets – particular larger ones like AIDA-CoNLL – can be (and are) used for training purposes.

[31] `http://www.cnts.ua.ac.be/conll2003/ner.tgz`

2.4.2. Third-party comparisons

We now describe third-party evaluations that compare various EEL tools. Note that we focus on evaluations that include a disambiguation step, and thus exclude studies that focus only on NER (e.g., [124]).

As previously discussed, Rizzo and Troncy [110] proposed the NERD approach to integrate various Entity Linking tools with online APIs. They also provided some comparative results for these tools, namely Alchemy, DBpedia Spotlight, Evri, OpenCalais and Zemanta [248]. More specifically, they compare the number of entities detected by each tool from 1,000 New York Times articles, considering six entity types: person, organization, country, city, time and number. These results show that while the commercial black box tools manage to detect thousands of entities, DBpedia Spotlight only detects 16 entities in total; to the best of our knowledge, the quality of the entities extracted was not evaluated. However, in follow-up work by Rizzo *et al.* [249], the authors use the AIDA–CoNLL dataset and a Twitter dataset to compare the linking precision, recall and F-measure of Alchemy, DataTXT, DBpedia Spotlight, Lupedia, TextRazor, THD, Yahoo! and Zemanta. In these experiments, Alchemy generally had the highest recall, DataTXT or TextRazor the highest precision, while TextRazor had the best F-measure for both datasets.

Gangemi [101] provides an evaluation of tools for Knowledge Extraction on the Semantic Web (or tools trivially adaptable to such a setting). Using a sample text obtained from an extract of an online article of The New York Times[32] as input, he evaluates the precision, recall, F-measure and accuracy of several tools for diverse tasks, including Named Entity Recognition, Entity Linking (referred to as Named Entity Resolution), Topic Detection, Sense Tagging, Terminology Extraction, Terminology Resolution, Relation Extraction, and Event Detection. Focusing on the EEL task, he evaluates nine tools: AIDA, Stanbol, CiceroLite, DBpedia Spotlight, FOX, FRED+Semiosearch, NERD, Semiosearch and Wikimeta. In these results, AIDA, CiceroLite and NERD have perfect precision (1.00), while Wikimeta has the highest recall (0.91); in a combined F-measure, Wikimeta fares best (0.80), with AIDA (0.78) and FOX (0.74) and CiceroLite (0.71) not far behind. On the other hand, the observed precision (0.75) and in particular recall (0.27) of DBpedia Spotlight was relatively low.

Cornolti *et al.* [57] presented an evaluation framework for Entity Linking systems, called the BAT-framework.[33] The authors use this framework to evaluate five systems – AIDA, DBpedia Spotlight, Illinois Wikifier, M&W Miner and TAGME (v2) – with respect to five publicly available datasets – AIDA–CoNLL, AQUAINT, IITB, Meij and MSNBC – that offer a mix of different types of inputs in terms of domains, lengths, densities of entity mentions, and so forth. In their experiments, quite consistently across the various datasets and configurations, AIDA tends to have the highest precision, TAGME and W&M Miner tend to have the highest recall, whilst TAGME tends to have the highest F-measure; one exception to this trend is the IITB dataset based on long webpages, where DBpedia Spotlight has the highest recall (0.50), while AIDA has very local recall (0.04); on the other hand, for this dataset, M&W Miner has the best F-measure (0.52). An interesting aspect of Cornolti *et al.*'s study is that it includes performance experiments, where the authors find that TAGME is an order of magnitude faster for the AIDA–CoNLL dataset than any other tool while still achieving the best F-measure on that dataset; on the other hand, AIDA and DBpedia Spotlight are amongst the slowest tools, being 2–3 orders of magnitude slower than TAGME.

Trani *et al.* [37] and Usbeck *et al.* [282] later provide evaluation frameworks based on the BAT-framework. First, Trani *et al.* proposed the DEXTER-EVAL, which allows to quickly load and run evaluations following the BAT framework.[34] Later, Usbeck *et al.* [282] proposed GERBIL[35], where the tasks defined for the BAT-framework are reused. GERBIL additionally packages six new tools (AGDISTIS, Babelfy, Dexter, NERD, KEA and WAT), six new datasets, and offers improved extensibility to facilitate the integration of new annotators, datasets, and measures. However, the focus of the paper is on the framework and although some results are presented as examples, they only involve particular systems or particular datasets.

Derczynski *et al.* [72] focused on a variety of tasks over tweets, including NER/EL, which has unique challenges in terms of having to process short texts with little context, heavy use of abbreviated mentions, lax capitalization and grammar, etc., but also has

[32] http://www.nytimes.com/2012/12/09/world/middleeast/syrian-rebels-tied-to-al-qaeda-play-key-role-in-war.html

[33] https://github.com/marcocor/bat-framework

[34] https://github.com/diegoceccarelli/dexter-eval

[35] http://aksw.org/Projects/GERBIL.html

unique opportunities for incorporating novel features, such as user or location modeling, tags, followers, and so forth. While a variety of approaches are evaluated for NER, with respect to EEL, the authors evaluate four systems – DBpedia Spotlight, TextRazor, YODIE and Zemanta – over two Twitter datasets – a custom dataset (where entity mentions are given to the system for disambiguation) and the Meij dataset (where the raw tweet is given). In general, the systems struggle in both experiments. YODIE – a system with adaptations for Twitter – performs best in the first disambiguation task (note that TextRazor was not tested). In the second task, DBpedia had the best recall (0.48), TextRazor had the highest precision (0.65) while Zemanta had the best F-measure (0.41) (note that YODIE was not run in this second test).

2.4.3. Challenge events

A variety of EEL-related challenge events have been co-located with conferences and workshops, providing a variety of standardized tasks and calling for participants to apply their techniques to the tasks in question and submit their results. These challenge events thus offer an interesting format for empirical comparison of different tools in this space. Amongst such events considering an EEL-related task, we can mention:

Entity Recognition and Disambiguation (ERD) is a challenge at the Special Interest Group on Information Retrieval Conference (SIGIR), where the ERD'14 challenge [34] featured two tasks for linking mentions to Freebase: a short-text track considering 500 training and 500 test keyword searches from a commercial engine, and a long-text track considering 100 training and 100 testing documents scraped from webpages.

Knowledge Base Population (KBP) is a track at the NIST Text Analysis Conference (TAC) with an Entity Linking Track, providing a variety of EEL-related tasks (including multi-lingual scenarios), as well as training corpora, validators and scorers for task performance.[36]

Making Sense of Microposts (#Microposts2016) is a workshop at the World Wide Web Conference (WWW) with a Named Entity rEcognition and Linking (NEEL) Challenge, providing a gold standard dataset for evaluating named entity recognition and linking tasks over microposts, such as found on Twitter.[37]

Open Knowledge Extraction (OKE) is a challenge hosted by the European Semantic Web Conference (ESWC), which typically contains two tasks, the first of which is an EEL task using the GERBIL framework [282]; ADEL [233] won in 2015 while WESTLAB [38] won in 2016.[38]

Workshop on Noisy User-generated Text (W-NUT) is hosted by the Annual Meeting of the Association for Computational Linguistics (ACL), which provides training and development data based on the ConLL data format.[39]

We highlight the diversity of conferences at which such events have been hosted – covering Linguistics, the Semantic Web, Natural Language Processing, Information Retrieval, and the Web – indicating the broad interest in topics relating to EEL.

2.5. Summary

Many EEL approaches have been proposed in the past 15 years or so – in a variety of communities – for matching entity mentions in a text with entity identifiers in a KB; we also notice that the popularity of such works increased immensely with the availability of Wikipedia and related KBs. Despite the diversity in proposed approaches, the EEL process is comprised of two conceptual steps: recognition and disambiguation.

In the recognition phase, entity mentions in the text are identified. In EEL scenarios, the dictionary will often play a central role in this phase, indexing the labels of entities in the KB as well as contextual information. Subsequently, mentions in the text referring to the dictionary can be identified using string-, token- or NER-based approaches, generating candidate links to KB identifiers. In the disambiguation phase, candidates are scored and/or selected for each mention; here, a wide range of features can be considered, relying on information extracted about the mention, the keywords in the context of the mentions and the candidates, the graph induced by the similarity and/or relatedness of mentions and candidates, the categories of an external reference corpus, or the linguistic dependen-

[36] `http://nlp.cs.rpi.edu/kbp/2014/`

[37] `http://microposts2016.seas.upenn.edu/challenge.html`

[38] `https://project-hobbit.eu/events/open-knowledge-extraction-oke-challenge-at-eswc-2017/`

[39] `http://noisy-text.github.io/2017/`

cies in the input text. These features can then be combined by various means – thresholds, objective functions, classifiers, etc. – to produce a final candidate for each mention or a support for each candidate.

With respect to the EEL task, given the breadth of approaches now available for this task, a challenging question is then: which EEL approach should I choose for application X? Different options are associated with different strengths and weaknesses, where we can highlight the following key considerations in terms of application requirements:

- *KB selection:* While some tools are general and accept or can be easily adapted to work with arbitrary KBs, other tools are more tightly coupled with a particular KB, relying on features inherent to that KB or a contextual source such as Wikipedia. Hence the selection of a particular target KB may already suggest the suitability of some tools over others.
- *Domain selection:* When working within a specific topical domain, the amount of entities to consider will often be limited. However, certain domains may involve types of entity mentions that are atypical; for example, while types such as persons, organizations, locations are well-recognized, considering the medical domain as an example, diseases or (branded) drugs may not be well recognized and may require special training or configuration.
- *Text characteristics:* Aside from the domain (be it specific or open), the nature of the text input can better suit one type of system over another. For example, even considering a fixed medical domain, Tweets mentioning illnesses will offer unique EEL challenges (short context, slang, lax capitalization, etc.) versus news articles, webpages or encyclopedic articles about diseases, where again, certain tools may be better suited for certain input text characteristics.
- *Language:* Along similar lines, most tools are designed or evaluated primarily around the English language, while others perform experiments for other common languages, such as Spanish, French, Chinese, etc. Language can be an important factor, where certain tools may rely on resources (stemmers, lemmatizers, POS-taggers, parsers, training datasets, etc.) that assume a particular language. Likewise, tools that do not use any language-specific resources may still rely to varying extents on features (such as capitalization, distinctive proper nouns, etc.) that will be present to varying extents in different languages.
- *Emerging entities*: As data changes upon time, new entities are constantly generated, which may require models to be periodically updated. An application may thus require at least the recognition of emerging entities, even if no link is provided to the KB or a NIL[40] identifier is provoked. On the other hand, even if an application does not require recognition of emerging entities when considering a given approach or tool, it may be important to consider the cost/feasibility of periodically updating the KB in dynamic scenarios (e.g., recognizing emerging trends in social media).
- *Output quality*: Quality is often defined as “fit for purpose”, where an EEL output fit for one application/purpose might be unfit for another. For example, a semi-supervised application where a human expert will later curate links might emphasize recall over the precision of the top-ranked candidate chosen, since rejecting erroneous candidates is faster than searching for new ones manually [70]; on the other hand, a completely automatic system may prefer a cautious output, prioritizing precision over recall. Likewise, some applications may only care if an entity is linked once in a text, while others may put a high priority on repeated (short) mentions also being linked. Different purposes provide different instantiations of the notion of quality, and thus may suggest the fitness of one tool over another.
- *Performance and overhead*: In scenarios where EEL must be applied over large and/or highly dynamic inputs, the performance of the EEL system becomes a critical consideration, where tools can vary in orders of magnitude with respect to runtimes. Likewise, EEL systems may have prohibitive hardware requirements, such as having to store the entire dictionary in primary memory, and/or the need to collectively model all mentions and entities in a given text in memory, etc. The requirements of a particular system can then be an important practical factor in certain scenarios.
- *Various other considerations*, such as availability of software, availability of appropriate training data, licensing of software, API restrictions, costs, etc., will also often apply.

[40] Not In Lexicon

Given that different approaches have their own inherent strengths and weaknesses, a number of ensemble systems (DEXTER 2, NERD, WESTLAB) have been proposed to run an input text over several EEL tools and to use a common model – such as an ontology – to integrate the results and to try to learn how best to combine the results of individual tools according to where and how they perform best. Such ensemble methods can often outperform the underlying individual tools, but at the cost of requiring suitable training data to learn how best to combine the results of these tools, as well as the performance cost of having to run EEL over several (in some cases computationally costly) tools, which may make ensemble approaches unsuitable in certain scenarios or applications.

Otherwise, to help users compare options, a variety of general evaluations and evaluation frameworks (BAT, DEXTER-EVAL, NERD, GERBIL) have emerged to compare the precision, recall, accuracy and/or performance of EEL tools for varying types of datasets and settings. In cases where a particular application requirement is not covered or tested by an existing evaluation, frameworks such as DEXTER-EVAL and GERBIL are pre-installed with a number of popular EEL systems and can be extended with new application-specific tasks and/or new systems.

In summary, modern EEL techniques are sufficiently mature for a variety of applications, as evidenced, for example, by the variety of commercial services and tools for EEL that have been made available. On the other hand, no one EEL system fits all and EEL remains an active area of research, with recent trends looking to apply joint inferencing techniques to solve interrelated problems in a unified manner, as well as various works looking at EEL in specific scenarios where domain-specific features can be exploited.

3. Concept Extraction & Linking

A given text collection may refer to one or one or several domains, such as Medicine, Finance, War, Technology, and so forth. Such domains may be associated with various concepts indicating a more specific topic, such as "`breast cancer`", "`solid state disks`", etc. Concepts (unlike entities) are often hierarchical, where, for example, "`breast cancer`", "`melanoma`", etc., may indicate concepts that specialize the more general concepts of "`cancer`", which in turn specializes the concept of "`disease`", etc.

For the purposes of this section, we coin the generic phrase "*Concept Extraction & Linking*" to encapsulate the following three related but subtly distinct Information Extraction tasks – previously discussed in Section A – that can be brought to bear in terms of gaining a greater understanding of the concepts spoken about in a text collection, which in turn can help, for example, to understand the important concepts in the domain that a collection of documents are about, or the topic of a particular document.

Terminology Extraction (TE): Given a text collection we know to be in a given domain, we may be interested to learn what terms/concepts are core to the terminology of that domain.[41] (See Listing 12, Appendix A, for an example.)

Keyphrase Extraction (KE): This task focuses on extracting important keyphrases for a given document.[42] In contrast with TE, which focuses on extracting important concepts relevant to a given domain, KE is focused on extracting important concepts relevant to a particular text. (See Listing 13, Appendix A, for an example.)

Topic Modeling (TM): The goal of Topic Modeling is to analyze cooccurrences of related keywords and cluster them into candidate grouping that potentially capture higher-level semantic "topics". (See Listing 14, Appendix A, for an example.)

There is a clear connection between TE and KE: though the goals are somewhat divergent – the former focuses on understanding the domain itself while the latter focuses on categorizing documents – both require extraction of domain terms/keyphrases from text and hence we summarize works in both areas together.

Likewise, the methods employed and the results gained through TE and KE may also overlap with the previously studied task of Entity Extraction & Linking (EEL). Abstractly, one can consider EEL as focusing on the extraction of individuals, such as "`Saturn`"; on the other hand, TE and KE focus on the extraction of conceptual terms, such as "`planets`". However, this distinction is often fuzzy, since TE and KE approaches may identify "`Saturn`" as a term referring to a domain concept, while EEL approaches may identify "`planets`" as an entity mention. Indeed, some papers that claim to perform keyphrase extraction are indistinguishable from techniques for performing entity extraction/linking [190], and vice versa.

[41] Also known as *term extraction* [90], *term recognition* [10], *vocabulary extraction* [77], *glossary extraction* [56], etc.

[42] Often simply referred to as *Keyword Extraction* [202,140]

However, we can draw some clear general distinctions between EEL and the domain extraction tasks discussed in this section: the goal in EEL is to extract all entities mentioned, while the goal in TE and KE is to extract a succinct set of domain-relevant keywords that capture the terminology of a domain or the subject of a document. When compared with EEL, another distinguishing aspect of TE, KE and TM is that while the former task will produce a flat list of candidate identifiers for entity mentions in a text, the latter tasks (often) go further and attempt to induce hierarchical relations or clusters from the extracted terminology.

In this section, we discuss works relating to TE, KE and TM that directly relate to the Semantic Web, be it to help in the process of building an ontology or KB, or using an ontology or KB to guide the extraction process, or linking the results of the extraction process to an ontology or KB. We highlight that this section covers a wide diversity of works from authors working in a wide variety of domains, with different perspectives, using different terminology; hence our goal is to cover the main themes and to abstract some common aspects of these works rather than to capture the full detail of all such heterogeneous approaches.

Example: From the term/keyphrase extraction examples provided in Listings 12 and 13 (Appendix A), one motivation for applying such techniques in the context of the Semantic Web is to link the extracted terms with disambiguated identifiers from a KB. An extract of such facts is provided in Listing 2, where `te` and `ke` denotes term extraction and keyword extraction elements respectively, and the output consists of (hypothetical) RDF triples linking extracted terms to categorical concepts described in the DBpedia KB. These linked categories in the KB are then associated with hierarchical relations that may be used to generalize or better understand the topic of the document.

Listing 2: Concept Extraction and Linking example

```
Input:  [Breaking Bad Wikipedia article text]

Output sample (TE/KE):
primetime emmy awards (te)
golden globe awards (te)
american crime drama television series (te, ke)
amc network (ke)

Output (CEL):
dbr:Breaking_Bad dct:subject
  dbc:Primetime_Emmy_Award_winners ,
  dbc:Golden_Globe_winners ,
  dbc:American_crime_drama_television_series ,
  dbc:AMC_(TV_channel)_network_shows .
```

Applications: In the context of the Semantic Web, a core application of such tasks – and a major focus of TE in particular – is to help with the creation, validation or extension of a domain ontology. Automatically extracting an expressive domain ontology from text is, of course, an inherently challenging task that falls within the area of *ontology learning* [173,212,48, 29,295]. In the context of TE, the focus is on extracting a *terminological ontology* [157,90] (aka. *lexicalized ontology* [214], *termino-ontology* [214] or *simple ontology* [40]), which captures terms referring to important concepts in the domain, potentially including a taxonomic hierarchy between concepts or identifying terms that are aliases for the same concept. The resulting concepts (and hierarchy) may be used, for example, in a semi-automated ontology building process to seed or extend the concepts the ontology models.[43]

Other applications relate to categorizing documents in a text collection according to their key concepts, and thus by topic and/or domain; this is the focus of the KE and TM tasks in particular. When these high-level topics are related back to a particular knowledge-base, this can enable various forms of *semantic search* [113, 161,275], for example to navigate the hierarchy of domains/topics represented by the KB while browsing or searching documents. Other applications include *text enrichment* or *semantic annotation* whereby terms in a text are tagged with structured information from a reference KB or ontology [288,150,210,287].

Process: The first step in all such tasks is the extraction of candidate domain terms/keywords in the text, which may be performed using variations on the methods for EEL; this process may also involve a reference ontology or knowledge-base used for dictionary or learning purposes, or to seed patterns. The second step is to perform a filtering of the terms, selecting only those that best reflect the concepts of the domain or the subject of a document. A third optional step is to induce a hierarchy or clustering of the extracted terms, which may lead to either a taxonomy or a topic model; in the case of a topic model, a further step may be to identify a singular term that identifies each cluster. A final optional step may be to link terms or topic identifiers to an existing KB, including disambiguation

[43] In the context of ontology building, some authors distinguish an *onomasiological* process from a *semasiological* process, where the former process relates to taking a known concept in an ontology and extracting the terms by which it may be referred to in a text, while the latter process involves taking terms and extracting their underlying conceptual meaning in the form of an ontology [33].

where necessary (if not already implicit in a previous step). In fact, the steps described in this process may not always be sequential; for example, where a reference KB or ontology is used, it may not be necessary to induce a hierarchy from the terms since such a hierarchy may already be given by the reference source.

System overview: Before discussing CEL techniques in more detail, we give an overview of the TE/KE/TM systems covered, including their overall purpose; the year of publication; and the main domain covered. Again, we only highlight systems that deal with the Semantic Web in a direct way, and that have a publication offering details. With respect to the domain reported, many of the techniques and methodologies mentioned here can be either adapted or applied directly to other domains, where we simply give the primary domain discussed in the relevant publication, typically also the domain for which evaluation results are presented. Also, in the case that a publication does not explicitly provide a name for their system/technique, we use the last name of the first author and the initials of the last name of the remaining authors as a "key" (such cases are italicized to distinguish them). Again, works are ordered first by year, then alphabetically.

OntoLearn (2003) [285,212] applies term extraction using WordNet as a reference to perform semantic interpretation of terms (meaning to select the correct sense in WordNet; i.e., WSD). A use-case is presented for automatically translating multi-word terms extracted from text in the tourism domain from English to Italian [212].

***MoriMIF* (2004) [202]** perform keyphrase extraction to capture the domains of interest of a person; relevant documents for the person are found through iterative Google searches. The person's interests are later modeled using the Friend of a Friend (FOAF) vocabulary[44].

OntoLT (2004) [30] uses term extraction to extend an existing ontology with concepts found in a domain-specific text collection. The system supports English and German and allows for extracting sub-class relations, as well as domain/range "slots" on properties. The tool is implemented as a plug-in for Protégé.

***CimianoHS* (2005) [49]** focus on the problem of extracting a hierarchy of terms using Formal Concept Analysis, from which a lattice of concepts is extracted and used to induce a hierarchy. The resulting system is tested over texts in the tourism and finance domains and compared with expert-defined taxonomies.

***GillamTA* (2005) [109]** perform term extraction over documents to help build domain ontologies, including detection of hyponyms and synonyms to build a taxonomy. Evaluation is conducted in the nanotechnology domain.

Text2Onto (2005) [173,52] is an Eclipse plug-in for extracting terms and thereafter building ontologies from text. A distinguishing feature of the approach is the assignment of probabilistic weights to the concepts and semantic relations produced, generating what is referred to as a "probabilistic ontology model" (POM).

***GullaBI* (2006) [113]** use lightweight techniques to perform term extraction over the internal documentation of a large petroleum company; the main use-case is to extract concepts and construct an ontology that can be used to enable semantic search over the given documentation.

***LemnitzerVKSECM* (2007) [161]** use keyword extraction to generate an ontology capturing relationships between query terms and relevant user concepts. The method is applied for eight language – Bulgarian, Czech, Dutch, English, German, Polish, Portuguese and Romanian – in the eLearning domain. Keywords are aligned to WordNet concepts and the upper ontology DOLCE for enabling semantic search.

***ChemudugunataHSS* (2008) [40]** combine concepts of an ontology with topics automatically generated for labeling collections of documents for the purposes of semantic annotation. In particular, an LDA topic model is used to extract novel topics/concepts from text that compliment ontological concepts during annotation. Experiments are conducted for educational texts in the domains of Science and Social Studies.

***JanikK* (2008) [139]** extract topics based on a custom RDF(S) KB extracted from Wikipedia. First entities are extracted from the document and linked with the KB. Then the graph induced by these entities is extracted and statistical analyses, including centrality measures, are used to select a topic label. Experiments are applied over news articles.

***DolbyFKSS* (2009) [77]** apply term extraction over documents, linking terms to types and categories in DBpedia, YAGO and Freebase. Evaluation is

[44] See `http://xmlns.com/foaf/0.1/`

conducted over Gartner analyst reports in the IT and Energy domains.

***ZhangYT* (2009) [303]** use ontologies to improve the term extraction process, where extracted terms are linked to and filtered by the ontology in question. Experiments are conducted over the Reuters-2158 text-classification dataset[45] using an ontology based on WordNet; however, the authors propose that their methods could extend to other ontologies, such as the Medical Subject Headings (MeSH) ontology.

CRCTOL (2010) [141] (*Concept–Relation–Concept Tuple-based Ontology Learning*) applies term extraction, inducing a concept hierarchy of domain-specific terms for the purposes of ontology learning. Experiments are provided over texts from the terrorism and sports domains.

***JainP* (2010) [138]** propose a topic modeling approach where a user first defines a domain-specific ontology; POS-based term extraction is then applied where users manually select relevant terms. These terms are linked to the ontology, where topic relevance is indicated by frequency of (sub-)terms in the document. Experiments are performed over educational documents from Computer Science.

PIRATES (2010) [238] applies keyphrase extraction to annotate documents with topical tags. The system uses a reference ontology to link and filter terms and to select high-level concepts as annotations. Experiments are conducted using an ontology from the Software Engineering domain.

TyDI (2010) [214] uses term extraction (powered by YaTeA [6]) in order to help build a domain ontology, where the system focuses on collaborative methods to refine and structure the initial vocabulary. A use-case is presented for building an ontology from biotechnology patents.

***MuñozGCHN* (2011) [206]** perform keyphrase extraction to identify the topic of social media posts, where keyphrases are mapped to DBpedia. Evaluation is conducted over social media posts from various platforms (social networks, forums, blogs, microblogs, etc.) in the telecommunications domain.

OSEE (2012) [149,150] uses term extraction to link terms from an ontology with mentions in a text. The system uses a variety of reference ontologies, including SwissProt, the Gene Ontology, the Gene Regulation Ontology, etc. The system is evaluated over MEDLINE abstracts where the goal is to semantically annotate gene terms and link them with proteins.

Canopy (2013) [133] first applies term extraction and LDA to generate an initial set of topics for a document; the terms are linked to DBpedia. In order to identify topics, the system extracts the KB sub-graph induced by the terms and then applies centrality measures, postulating high-scoring terms as suitable topic labels.

***CardilloWRJVS* (2013) [33]** perform term extraction over medical texts following various reference sources; terms are linked (where possible) to SNOMED-CT. Experiments were conducted in the domain of Cardiology, where a resulting multilingual terminology is published as Linked Data using the Lemon lexicon model [50].

F-STEP (2013) [184] performs term extraction in open domain settings with an emphasis on creating a taxonomy of terms relevant to a text collection. WikiMiner [197] is used for term extraction, with the results linked to a KB (DBpedia and Freebase are used). Through this linking process, a SKOS [194] taxonomy is induced.

Distiller (2014) [210] uses keywords to semantically annotate documents, linking keywords to DBpedia. Evaluation is conducted in an open domain, where the results are mapped to OpenGraph and Schema.org vocabularies.

FGKBTE (2014) [90] (*FunGram FB Term Extraction*) applies term extraction over documents in order to help build domain ontologies. Given a custom core KB – called FunGramKB – the goal is to create "satellite ontologies" capturing specific domains. Evaluation is conducted in the crime/terrorism domain.

***VargaCRCH* (2014) [284]** focus on topic classification for microposts (e.g., Tweets), where they first apply entity extraction using existing tools, which are linked to DBpedia and Freebase. Thereafter, these entities are enriched with KB relations (including categories, types, etc.) to produce a graph from which features are extracted and input into a SVM classifier to identify topics.

***AllahyariK* (2015) [4]** extract topics for a document where, initially, bi-grams are extracted as terms, filtered by frequency, enriched with text from Wikipedia, and linked to DBpedia. Then they apply a variation of LDA considering concepts as another latent variable. The sub-graph of DBpe-

[45] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

dia induced by the terms is extracted and HITS authority is applied to identify a topic label.

***ChenJYYZ* (2016) [43]** apply entity linking using the DBpedia Spotlight tool, where they then take the sub-graph of the KB induced by the entities, over which they apply weights and pLSA to model topics. Experiments are performed over Computer Science texts and news-group articles.

***LauscherNRP* (2016) [158]** apply entity-based topic modeling where entities are extracted, filtered by TF–IDF, and linked to DBpedia. The labels of these entities are then input into Labeled LDA to produce labeled topics. Experiments are conducted on European Parliament documents.

LiTeWi (2016) [56] performs term extraction over educational textbooks. Terms are mapped to Wikipedia articles using Wikiminer [197], which is also used to filter for domain relevance. Evaluation is conducted over textbooks relating to Programming, Astronomy and Molecular Biology.

***LossioJRT* (2016) [287]** perform term extraction in the Biomedical domain with respect to a variety of domain-specific ontologies using an ensemble of linguistic and statistical methods. Web-search is used to measure relatedness of terms. Evaluation is performed for English, French and Spanish text using the UMLS and SNOMED ontologies.

***TodorLAP* (2016) [277]** apply an entity-based topic modeling to documents, where entities are extracted and linked to DBpedia and enriched with various meta-data from the KB. The enriched entities are fed into an LDA process to generate a topic model for a given document. Experiments are conducted in the news domain.

An overview of the highlighted domain extraction systems is provided in Table 3, listing features we will elaborate upon in the following subsections.

Black box systems: Black-box semantic-annotation tools – such as AlchemyAPI, Apache Stanbol, OpenCalais, Textrazor, Zemanta, amongst others – likewise support features relating to the extraction of domain terms and their association with text documents. Discussion of such systems is out of scope since details of their implementation have not been published.

3.1. Term/Keyphrase Recognition

We consider a term to be a textual mention of a domain-specific concept, such as "`cancer`". Terms may also be composed of more than one word and indeed by relatively complex phrases, such as "`inner planets of the solar system`". Terminology then refers more generically to the collection of terms or specialized vocabulary pertinent to a particular domain. In the context of TE, terms/terminology are typically understood as describing a domain, while in the context of KE, keyphrases are typically understood as describing a particular document. However, the extraction process of TE and KE are similar; hence we proceed by generically discussing the extraction of terms.

In fact, approaches to extract raw candidate terms follows a similar approach to that for extracting raw entity mentions in the context of Entity Linking. Generic preprocessing methods such as stop-word removal, stemming and/or lemmatization are often applied, along with tokenization. Some term extraction methods then rely on window-based methods, extracting *n*-grams up to a predefined length [90]. Other term extractions apply POS-tagging and then define shallow syntactic patterns to capture, for example, noun phrases ("`solar system`"), noun phrases preceded by adjectives ("`inner planets`"), and so forth [206,77,56]. Other systems may use an ensemble of methods to extract a broad selection of terms that will be filtered in a subsequent step [56].

There are, however, subtle differences when compared with extracting entities, particularly when considering traditional NER scenarios looking for names of people, organizations, places, etc.; when extracting terms, for example, capitalization becomes less useful as a signal, and syntactic patterns may need to be more complex to identify concepts such as "`inner planets of [the] solar system`". Furthermore, the features considered when filtering term candidates change significantly when compared with those for filtering entity mentions (as we will discuss presently). For these reasons, various systems have been proposed specifically for extracting terms, including KEA [294], TExSIS [172], TermeX [69] and YaTeA [6] (here taking a selection reused by the highlighted systems).

3.2. Filtering

Once a set of candidate terms have been identified, a range of features can be used for either automatic or semi-automatic filtering. These features can be broadly categorized as being *linguistic* or *statistical*; however, other *contextual* features can be used, which will be described presently. Furthermore, filtering can be applied with respect to a domain-specific dictionary of terms as taken from a reference KB or ontology.

Table 3

Overview of Concept Extraction & Linking systems

Setting denotes the primary domain in which experiments are conducted; **Goal** indicates the stated aim of the system (KE: Keyphrase Extraction, OB: Ontology Building, SA: Semantic Annotation, TE: Terminology Extraction, TM: Topic Modeling); **Recognition** summarizes the term extraction method used; **Filtering** summarizes methods used to select suitable domain terms; **Relations** indicates the semantics relations extracted between terms in the text (Hyp.: Hyponyms, Syn.: Synonyms, Mer.: Meronyms); **Linking** indicates KBs/ontologies to which terms are linked; **Reference** indicates other sources used; **Topic** indicates the method(s) used to to model (and label) topics. '—' denotes no information found, not used or not applicable.

Name	**Year**	**Setting**	**Goal**	**Extraction**	**Filtering**	**Relations**	**Linking**	**Reference**	**Topic**
AllahyariK [4]	2015	Multi-domain	TM	Token-based	Statistical	—	DBpedia	Wikipedia	LDA / Graph
Canopy [133]	2013	Multi-domain	TM	—	—	—	DBpedia	Wikipedia	LDA / Graph
CardilloWRJVS [33]	2013	Medicine	TE	TExSIS [172]	Manual	—	SNOMED-CT DBpedia	—	—
ChemuduguntaHSS [40]	2008	Science Social Studies	SA	2008	—	Statistical	—	CIDE, ODP	LDA
ChenJYYZ [43]	2016	Comp. Sci. News	TM	DBpedia Spotlight	Graph/Stat.	—	DBpedia	—	pLSA / Graph
CimianoHS [49]	2005	Tourism Finance	OB	POS-based	Statistical	Hyp.	—	—	—
CRCTOL [141]	2010	Terrorism Sports	OB	POS-based	Statistical	Hyp.	—	WordNet	—
Distiller [210]	2014	—	KE	—	Hybrid	—	DBpedia	—	—
DolbyFKSS [77]	2009	Info. Tech. Energy	TE	POS-based	Statistical	—	DBpedia Freebase	—	—
F-STEP [184]	2013	News	TE	WikiMiner [197]	WikiMiner [197]	Hyp.	DBpedia Freebase	Wikipedia	—
FGKBTE [90]	2014	Crime Terrorism	TE	Token-based	Statistical	—	FunGramKB	—	—
GillamTA [109]	2005	Nanotechnology	OB	—	Hybrid	Hyp.	—	—	—
GullaBI [113]	2006	Petroleum	OB	POS-based	Statistical	—	—	—	—
JainP [138]	2010	Comp. Sci.	TM	POS-based	Stat. / Manual	—	Custom onto.	—	Hierarchy
JanikK [139]	2008	News	TM	—	Statistical	—	Wikipedia	—	Graph
LauscherNRP [158]	2016	Politics	TM	DBpedia Spotlight	Statistical	—	DBpedia	—	L-LDA
LemnitzerVKSECM [161]	2007	E-Learning	OB	POS-based	Statistical	Hyp.	OntoWordNet	Web search	—
LiTeWi [56]	2016	Programming Astronomy Biology	OB	Ensemble	Wikiminer [197]	—	Wikipedia	—	—
LossioJRT [287]	2016	Biomedical	TE	POS-based	Statistical	—	UMLS SNOMED-CT	Web search	—
MoriMIF [202]	2004	Social Data	KE	TermeX [69]	Statistical	—	—	Google	—
MuñozGCHN [206]	2011	Telecoms	KE	POS-based	Hybrid	—	DBpedia	Wikipedia	—
OntoLearn [285,212]	2001	Tourism	OB	POS-based	Statistical	Various	—	WordNet, Google	—
OntoLT [30]	2004	Neurology	OB	POS-based	Statistical	Hyp.	Any	—	—
OSEE [149,150]	2012	Bioinformatics	SA	POS-based	KB-based	—	Gene Ontology	Various (OBO)	—
PIRATES [238]	2010	Software	SA	KEA [294]	Hybrid	—	SEOntology	—	—
Text2Onto [173,52]	2005	—	OB	POS/JAPE	Statistical	Hyp. Mer.	—	WordNet	—
TodorLAP [277]	2016	News	TM	DBpedia Spotlight	—	Hyp.	DBpedia	Wikipedia	LDA
TyDI [214]	2010	Biotechnology	OB	—	YaTeA [6]	Syn. Hyp.	—	—	—
VargaCRCH [284]	2014	Microposts	TM	OpenCalais Zemanta	—	—	DBpedia Freebase	—	Graph / ML
ZhangYT [303]	2009	News	TE	Manual	Statistical	—	—	WordNet	—

Linguistic features relate to lexical or syntactic aspects of the term itself, where the most basic such feature would be the number of words forming the term (more words indicating more specific terms and vice-versa). Other linguistic features can likewise include generic aspects such as POS tags [206,202,113,161, 210], shallow syntactic patterns [202,109,77,56], etc.; such features may be used in the initial extraction of terms or as a post-filtering step. Furthermore, terms may be filtered or selected based on appearing in a particular hierarchical branch of terminology, such as terms relating to forms of cancer; these techniques will be discussed in the next subsection.

Statistical measures look more broadly at the usage of a particular term in a text collection. In terms of such measures, two key properties of terms are often analyzed in this context [56]: *unithood* and *termhood*.

Unithood refers to the cohesiveness of the term as referring to a single concept, which is often assessed through analysis of *collocations*: expressions where the meaning of each individual word may vary widely from their meaning in the expression such that the meaning of the word depends directly on the expression; an example collocation might be "`mean squared error`" where, in particular, the individual words "`mean`" and "`squared`" taken in isolation have meanings unrelated to the expression, where the phrase "`mean squared error`" thus has high unithood. There are then a variety of measures to detect collocations, most based on the idea of comparing the expected number of times the collocation would be found if occurrences of the individual words were independent versus the amount of times the collocation actually appears [80]. As an example, Lossio-Ventura *et al.* [287] use Web search engines to determine collocations, where they estimate the unithood of a candidate term as the ratio of search results returned for the exact phrase ("`mean squared error`"), versus the number of results returned for all three terms (`mean AND squared AND error`).

The second form of statistical measure, called *termhood*, refers to the relevance of the term to the domain in question. To measure termhood, variations on the theme of the TF–IDF measure are commonly used [113,161,77,90,56,287], where, for example, terms that appear often in a (domain) specific text (high TF) but appear less often in a general corpus (high IDF) indicate higher termhood. Note that termhood relates closely to topic extraction measures, where the context of terms is used to find topically-related terms; such approaches will be discussed later.

Other features can rather be contextual, looking at the position of terms in the text [238]; such features are particularly important in the context of identifying keyphrases/terms that capture the domain or topic of a given document. The first such feature is known as the *phrase depth*, which measures how early in the document is the first appearance of the term: phrases that appear early on (e.g., in the title or first paragraph) are deemed to be most relevant to the document or the domain it describes. Likewise, terms that appear throughout the entire document are considered more relevant: hence the *phrase lifespan* – the ratio of the document lying between the first and last occurrence of the term – is also considered as an important feature [238].

A KB can also be used to filter terms through a linking process. The most simple such procedure is to filter terms that cannot be linked to the KB [150]. Other proposed methods rather apply a graph-based filtering, where terms are first linked to the KB and then the subgraph of the KB induced by the terms is extracted; subsequently, terms in the graph that are disconnected [43] or exhibiting low centrality [133] can be filtered. This process will be described in more detail later.

3.3. Hierarchy Induction

Often the extracted terms will refer to concepts with some semantic relations that are themselves useful to model as part of the process. The semantic relations most often considered are synonymy (e.g., "`heart attack`" and "`myocardial infarction`" being synonyms) and hypernymy/hyponymy (e.g., "`migraine`" is a hyponym of "`headache`" with the former being a more specific form of the latter, while conversely "`headache`" is a hypernym of "`migraine`"). While synonymy induces groups of terms with (almost exactly) the same meaning, hypernymy/hyponymy induces a hierarchical structure over the terms. Such relations and structures can be extracted either by analysis of the text itself and/or through the information gained through some reference source. These relations can then be used to filter relevant terminology, or to induce an initial semantic structure that can be formalized as a taxonomy (e.g., expressed in the SKOS standard [194]), or as a formal ontology (e.g., expressed in the OWL standard [125]), and so forth. As such, this topic relates heavily to the area of ontology learning, where we refer to the textbook by Cimiano [46] and the more recent survey of Wong *et al.* [295] for details; here our goal is to capture the main ideas.

In terms of detecting hypernymy from the text itself, a key method relies on distinguishing the *head term*, which signifies the more general hypernym in a (potentially) multi-word term; from *modifier terms*, which then specialize the hypernym [285,30,141, 214]. For example, the head term of "`metastatic breast cancer`" is "`cancer`", while "`breast`" and "`metastatic`" are modifiers that specialize the head term and successively create hyponyms. As a more complex example, the head term of "`inner planets of the solar system`" would be "`planets`", while "`inner`" and "`of the solar system`" are modifying phrases. Analysis of the head/modifier terms thus allows for automatic extraction of hypernymic relations, starting with the most general head term, such as "`cancer`", and then subsequently extending to hyponyms by successively adding modifiers appearing in a given multi-word terms, such as "`breast cancer`", "`metastatic breast cancer`", and so forth.

Of course, analyzing head/modifier terms will miss hypernyms not involving modifiers, and synonyms; for example, the hyponym "`carcinoma`" of "`cancer`" is unlikely to be revealed by such analysis. An alternative approach is to rely on lexico-syntactic patterns to detect synonymy or hypernymy. A common set of patterns to detect hypernymy are *Hearst patterns* [119], which look for certain connectives between noun phrases. As an example, such patterns may detect from the phrase "`cancers, such as carcinomas, ...`" that "`carcinoma`" is a hyponym of "`cancer`". Hearst-like patterns are then used by a variety of systems (e.g., [52,109,141,150]). While such patterns can capture additional hypernyms with high precision [119], Buitelaar *et al.* [29] note that finding such patterns in practice is rare and that the approach tends to offer low recall. Hence, approaches have been proposed to use the vast textual information of the Web to find instances of such patterns using, for example, web search engines such as Google [47], the abstracts of Wikipedia articles [277], etc.

Another approach that potentially offers higher recall is to use statistical analyses of large corpora of text. Many such approaches are based, for example, on distributional semantics, which aggregates the context (surrounding terms) in which a given term appears in a large text corpus. The distributional hypothesis then considers that terms with similar contexts are semantically related. Within this grouping of approaches, one can then find more specific strategies based on various forms of clustering [48], Formal Concept Analysis [49], and more recently LDA [54], etc., to find and induce a hierarchy from terms with similar contexts, or more/less specific contexts. These can then be used as the basis to detect synonyms; or more often to induce a hierarchy of hypernyms, possibly adding *hidden concepts* – fresh hypernyms of cohyponyms – to create a connected tree of more/less specific domain terms.

Of course, reference resources that already contain semantic relations between terms can be used to aid in this process. One important such resource is WordNet [195], which, for a given term, provides a set of possible semantic senses in terms of what it might mean (homonymy/polysemy [283]), as well as a set of synonyms called *synsets*. Those synsets are then related by various semantic relations, including hypernymy, meronymy (part of), etc. WordNet is thus a useful reference for understanding the semantic relations between concepts, used by a variety of systems (e.g., [212,52,303,141], etc.). Other systems rather rely on, for example, Wikipedia categorizations [184] in combination with reference KBs. A core challenge, however, when using such approaches is the problem of *word sense disambiguation* [211] (sometimes called the *semantic interpretation* problem [212]): given a term, determine the correct sense in which it is used. We refer to the survey by Navigli [212] for discussion.

While such approaches can induce an initial hierarchy of terms, it is important to note that this is a challenging task. As, for example, Blomqvist [23] notes, the quality of the output of such methods is "far from perfect": the result will often contain large sets of disconnected concepts (forming a forest rather than a tree), be relatively sparse (not containing many relations), and be quite shallow. Thus the output of such techniques should be seen as something to seed or complement an ontology building process rather than being able to produce a final high-quality domain ontology/taxonomy in a fully-automated manner.

An alternative to extracting semantic relations between terms in the text is to instead rely on the existing relations in a given KB. That is to say, if the terms can be linked to a suitable existing KB, then semantic relations can be extracted from the KB itself rather than the text. This approach is often applied by tools described in the following section (e.g., [133,284,43]) whose goal is to understand the domain of a document rather than attempting to model a domain from text.

3.4. Topic Extraction

While the previous methods are mostly concerned with extracting a terminology from a text collection that describes a given domain (e.g., for the purposes of building a domain-specific ontology), other works are concerned with modeling and potentially identifying the domain to which the documents in a given text collection pertain (e.g., for the purposes of classifying documents). We refer to these latter approaches generically as *topic extraction* approaches [165,185].[46] Such approaches are based on analysis of terms (or sometimes entities) extracted from the text over which topic modeling approaches can be applied to cluster and analyze thematically related terms (e.g., "`carcinoma`", "`malignant tumour`", "`chemotherapy`"). Thereafter, *topic labeling* [133,4] can be (optionally) applied to assign such grouping of terms a suitable KB identifier referring to the topic in question (e.g., `dbr:Cancer`). Application areas for such techniques include Information Retrieval [227], Recommender Systems [143], Text Classification [132], Cognitive Science [230], and Social Network Analysis [244], to name but a few.

For applying topic modeling, one can of course first consider directly applying the traditional methods proposed in the literature: LSA, pLSA and/or LDA (see Appendix A for discussion). However, these approaches have a number of drawbacks. First, such approaches typically work on individual words and not multi-word terms (though extensions have been proposed to consider multi-word terms). Second, topics are considered as latent variables associated with a probability of generating words, and thus are not directly "labeled", making them difficult to explain or externalize (though, again, labeled extensions have also been proposed, for example for LDA). Third, words are never semantically interpreted in such models, but are rather considered as symbolic references over which statistical/probabilistic inference can be applied. Hence a number of approaches have emerged that propose to use the structured information available in KBs and/or ontologies to enhance the modeling of topics in text. The starting point for all such approaches is to extract some terms from the text, using approaches previously outlined: some rely simply on token- or POS-based methods to extract terms, which can be filtered by frequency or TF–IDF variants to capture domain relevance [139,138,4], whereas others rather prefer entity recognition tools (which are subsequently mapped to higher-level topics through relations in the KB, as we describe later) [43,158,277].

With extracted terms in hand, the next step for many approaches – departing from traditional topic modeling – is to link those terms to a given KB, where the semantic relations of the KB can be exploited to generate more meaningful topics. The most straightforward such approach is to assume an ontology that offers a concept/class hierarchy to which extracted terms from the document are mapped. Thus the ontology can be seen as guiding the topic modeling process, and in fact can be used to select a label for the topic. One such approach is to apply a statistical analysis over the term-to-concept mapping. For example, in such a setting, Jain and Pareek [138] propose the following: for each concept in the ontology, count the ratio of extracted terms mapped to it or its (transitive) sub-concepts, and take that ratio as an indication of the relevance of the concept in terms of representing a high-level topic of the document. Another approach is to consider the spanning tree(s) induced by the linked terms in the hierarchy, taking the lowest common ancestor(s) as a high-level topic [133]. However, as noted by Hulpuş *et al.* [133], such approaches relying on class hierarchies tend to elect very generic topic labels, where they give the example of "`Barack Obama`" being captured under the generic concept *person*, rather than a more interesting concept such as *U.S. President*. To tackle this problem, a number of approaches have proposed to link terms – including entity mentions – to Wikipedia's categories, from which more fine-grained topic labels can be selected for a given text [258,135,60].

Other approaches apply traditional topic modeling methods (typically pLSA or LDA) in conjunction with information extracted from the KB. Some approaches propose to apply topic modeling in an initial phase directly over the text; for example, Canopy [133] applies LDA over the input documents to group words into topics and then subsequently links those words with DBpedia for labeling each topic (described later). On the other hand, other approaches apply topic modeling after initially linking terms to the KB; for example, Todor *et al.* [277] first link terms to DBpedia in order to enrich the text with annotations of types, categories, hypernyms, etc., where the enriched text is then passed through an LDA process. Some recent approaches rather extend traditional topic models to consider information from the KB *during* the inference of topic-related distributions. Along these lines, for example, Allahyari [4] propose an LDA variant, called *OntoLDA*, which introduces a latent variable for concepts (taken from DBpedia and linked with the text), which sits between words and topics: a document is

[46] Also known as *topic classification* [284].

then considered to contain a distribution of (latent) topics, which contains a distribution of (latent) concepts, which contains a distribution of (observable) words. Another such hybrid model, but rather based on pLSA, is proposed by Chen *et al.* [43] where the probability of a concept mention (or a specific entity mention[47]) being generated by a topic is computed based on the distribution of topics in which the concept or entity appears and, more importantly, the same probability for entities that are related in the KB (with a given weight).

The result of these previous methods – applying topic modeling in conjunction with a KB-linking phase – is a set of topics associated with a set of terms that are in turn linked with concepts/entities in the KB. Interestingly, the links to the KB then facilitate labeling each topic by selecting one (or few) core term(s) that help capture or explain the topic. More specifically, a number of graph-based approaches have recently been proposed to choose topic labels [139,133,4], which typically begin by selecting, for each topic, the nodes in the KB that are linked by terms under that topic, and then extracting a sub-graph of the KB in the neighborhood of those nodes, where typically the largest connected component is considered to be the topical/thematic graph [139,4]. The goal, thereafter, is to select the "label node(s)" that best summarize(s) the topic, for which a number of approaches apply centrality measures on the topical graph: Janik and Kochut [139] investigate use of a closeness centrality measure, Allahyari and Kochut [4] propose to use the authority score of HITS (later mapping central nodes to DBpedia categories), while Hulpuş *et al.* [133] investigate various centrality measures, including closeness, betweenness, information and random-walk variants, as well as "focused" centrality measures that assign special weights to nodes in the topic (not just in the neighborhood). On the other hand, rather than applying traditional topic modeling algorithms, Varga *et al.* [284] propose to extract a KB sub-graph (from DBpedia or Freebase) describing entities linked from the text (containing information about classes, properties and categories), over which weighting schemes are used to derive a set of input features for a machine learning model (SVM) that classifies the topic of microposts.

3.5. Representation

Domain knowledge extracted through the previous processes may be represented using a variety of Semantic Web formats. In the ontology building process, induced concepts may be exported to RDFS/OWL [125] for further reasoning tasks or ontology refinement and development. However, RDFS/OWL makes a distinction between concepts and individuals that may be inappropriate for certain modeling requirements; for example, while a term such as "`US Presidents`" could be considered as a sub-topic of "`US Politics`" since any document about the former could be considered also as part of the latter, the former is neither a sub-concept nor an instance of the latter in the set-theoretic setting of OWL.[48] For such scenarios, the Simple Knowledge Organization System (SKOS) [194] was standardized for modeling more general forms of conceptual hierarchies, taxonomies, thesauri, etc., including semantic relations such as broader-than (e.g., hypernym-of), narrow-than (e.g., hyponym-of), exact-match (e.g., synonym-of), close-match (e.g., near-synonym-of), related (e.g., within-same-topic-as), etc.; the standard also offers properties to define primary labels and aliases for concepts.

Aside from these Semantic Web standards, a number of other representational formats have been proposed in the literature. The Lexicon Model for Ontologies (Lemon) [50] was proposed as a format to associate ontological concepts with richer linguistic information, which, on a high level, can be seen as a model that bridges between the world of formal ontologies to the world of natural language (written, spoken, etc.); the core Lemon concept is a lexical entry, which can be a word, affix or phrase (e.g., "`cancer`"); each lexical entry can have different forms (e.g., "`cancers`", "`cancerous`"), and can have multiple senses (e.g., "`ex:cancer_sense1`" for medicine, "`ex:cancer_sense2`" for astrology, etc.); both lexical entries and senses can then be linked to their corresponding ontological concepts (or individuals).

Along related lines, Hellmann *et al.* [120] propose the NLP Interchange Format (NIF), whose goal is to enhance the interoperability of NLP tools by using an ontology to describe common terms and concepts; the format can provide Linked Data as output for further

[47] They use the term *entity* to refer to both concepts, such as *person*, and individuals, such as *Barack Obama*.

[48] It is worth noting that OWL does provide means for *meta-modeling* (aka. punning), where concepts can be simultaneous considered as groups of individuals when reasoning at a terminological level, and as individuals when reasoning at an assertional level.

data re-use. In fact, a variety of proposals have also been made in terms of publishing linguistic resources as Linked Data. Cimiano *et al.* [51] propose such an approach for publishing and linking terminological resources following the Linked Data principles, combining the Lemon, SKOS, and PROV-O vocabularies in their core model; OnLit was proposed by Klimek et al [154] as a Linked Data version of the LiDo Glossary of Linguistic Terms; etc. For further information, we refer the reader to the editorial by McCrae *et al.* [182], which offers an interesting diagram of terminological/linguistic resources published as Linked Data.

3.6. Evaluation

Given the diversity of approaches gathered together in this section, we remark that the evaluation strategies employed are likewise equally diverse. In particular, evaluation varies depending on the particular task considered (be it TE, KE, TM or some combination thereof) and the particular application in mind (be it ontology building, text classification, etc.). Evaluation in such contexts is often further complicated by the potentially subjective nature of the goal of such approaches. When assessing the quality of the output, some questions may be straightforward to answer, such as: *Is this phrase a singular term (unithood/precision)?* On the other hand, evaluation must somehow deal with more subjective domain-related questions, such as: *Is this a domain-relevant term (termhood/precision)? Have we captured all domain-relevant terms appearing in the text (recall)? Is this taxonomy of terms correct (precision)? Does this label represent the terms forming this topic (precision)? Does this document have these topics (precision)? Are all topics of the document captured (recall)?* And so forth. Such questions are inherently subjective, may raise disagreement amongst human evaluators [161], and may require expertise in the given domain to answer adequately.

Datasets: For evaluating such approaches, notably there are many Web-based corpora that have been pre-classified with topics or keywords, often annotated by human experts – such as users, moderators or curators of a particular site – through, for example, tagging systems. These can be reused for evaluation of domain extraction tools, in particular to see if automated approaches can recreate the high-quality classifications or topic models inherent in such corpora. Some such corpora that have been used include: BBC News[49] [133,277], British Academic Written Corpus[50] [133,4], British National Corpus[51] [49], CNN News [139], DBLP[52] [43], eBay[53] [288], Enron Emails[54], Twenty Newsgroups[55] [43,277], Reuters News[56] [162,49,303], StackExchange[57] [133], Web News[58] [277], Wikipedia categories [60], Yahoo! categories [276], and so forth. Other datasets have been specifically created as benchmarks for such approaches. In the context of KE, for example, gold standards such as SemEval [151] and Crowd500 [177] have been created, with the latter being produced through a crowdsourcing process; such datasets were used by Gagnon *et al.* [100] and Jean-Louis *et al.* [140] for KE-related third-party evaluations. In the context of TE, existing domain ontologies can be used – or manually created and linked with text – to serve as a gold standard for evaluation [30,49,303,214,150].

Rather than employing a pre-annotated gold standard, an alternative strategy is to apply the approach under evaluation to a non-annotated text collection and thereafter seek human judgment on the output, typically in comparison with baseline approaches from the literature. Such an approach is employed in the context of TE by Nédellec *et al.* [214], Kim and Tuan [150], and Dolby *et al.* [77]; or in the context of KE by Muñoz-García *et al.* [206]; or in the context of TM by Hulpuş *et al.* [133] and Lauscher *et al.* [158]; and so forth. In such evaluations, TM approaches are often compared against traditional approaches such as LDA [4], PLSA [43], hierarchical clustering [116], etc.

Metrics: The most commonly used metrics are *precision*, *recall*, and *F_1 measure*. However, other metrics can be used to assess the quality of the output of particular tasks or components. For example, evaluations in the area of topic modeling may also consider *perplexity* (the log-likelihood of a held-out test set) and/or

[49] http://mlg.ucd.ie/datasets/bbc.html

[50] http://www2.warwick.ac.uk/fac/soc/al/research/collections/bawe/

[51] http://www.natcorp.ox.ac.uk/

[52] http://dblp.uni-trier.de/db/

[53] http://www.ebay.com

[54] https://www.cs.cmu.edu/~./enron/

[55] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

[56] http://www.daviddlewis.com/resources/testcollections/reuters21578/ and http://www.daviddlewis.com/resources/testcollections/rcv1/

[57] https://archive.org/details/stackexchange

[58] http://mklab.iti.gr/project/web-news-article-dataset

coherence [250], where a topic is considered coherent if it covers a high ratio of the words/terms appearing in the textual context from which it was extracted (measured by metrics such as Pointwise Mutual Information (PMI) [56], Normalized Mutual Information (NMI), etc.). When comparing with baseline approaches over *a priori* non-annotated corpora, recall can be a manually-costly aspect to assess; hence comparative rather than absolute measures such as *relative recall* are sometimes used [150]. In cases where results are ranked, precision@k measures may be used to assess the quality of the top-k terms extracted [287].

3.7. Summary

In this section, we gather together three approaches for extracting domain-related concepts from a text: Term Extraction (TE), Keyphrase Extraction (KE), and Topic Modeling (TM). While the first task is typically concerned with applications involving ontology building, or otherwise extracting a domain-specific terminology from an appropriate text collection, the latter two tasks are typically concerned with understanding the domain of a given text. As we have seen in this section, all tasks relate in important aspects, particularly in the identification of domain-relevant concepts; indeed, TE and TM further share the goal of extracting relationships between the extracted terms, be it to induce a hierarchy of hypernyms, or find synonyms, or find thematically-related clusters of terms.

While all of the discussed approaches rely – to varying degrees – on techniques proposed in the traditional IE/NLP literature for such tasks, the use of reference ontologies and/or KBs has proven useful for a number of technical aspects inherent to these tasks:

- using the entity labels and aliases of a domain-specific KB as a dictionary to guide the extraction of conceptual domain terms in a text [150,287];
- linking terms to KB concepts and using the semantic relations in the KB to determine (un)related terms [139,133,184,4,43];
- enriching text with additional information taken from the KB [277];
- classifying text with respect to an ontological concept hierarchy [138];
- building topic models that include semantic relations from the KB [4,43];
- determining topic labels/identifiers based on centrality of nodes in the KB graph [133,4,43];
- representation and integration of terminological knowledge [194,50,120,33,51,182].

On the other hand, such processes are also often used to create or otherwise enrich Semantic Web resources, such as for ontology building applications, where TE can be used to extract a set of domain-relevant concepts – and possibly some semantic relations between them – to either seed or enhance the creation of a domain-specific ontology [285,212,48,52,46,295].

4. Relation Extraction & Linking

At the heart of any Semantic Web KB are relations between entities [261]. Thus an important traditional IE task in the context of the Semantic Web is *Relation Extraction* (RE), which is the process of finding relationships between entities in the text. Unlike the tasks of Term Extraction or Topic Modeling that aim to extract fixed relationships between concepts (e.g., hypernymy, synonymy, relatedness, etc.), RE aims to extract instances of a broader range of relations between entities (e.g., born-in, married-to, interacts-with, etc.). Relations extracted may be binary relations or even higher arity n-ary relations. When the predicate of the relation – and the entities involved in it – are linked to a KB, or when appropriate identifiers are created for the predicate and entities, the results can be used to (further) populate the KB with new facts. However, first it is also necessary to represent the output of the relation extraction process as RDF: while binary relations can be represented directly as triples, n-ary relations require some form of reified model to encode.

In this section, we thus discuss approaches for extracting relations from text and linking their constituent predicates (and entities) to an ontology/KB; we also discuss representations used to encode results as RDF in support of the KB population task.

Example: In Listing 3, we provide a hypothetical (and rather optimistic) example of Relation Extraction and Linking with respect to DBpedia. Given a textual statement, the output provides an RDF representation of entities interacting through relationships associated with properties of an ontology. Note that there may be further information in the input not represented in the output, such as that the entity "`Bryan Lee Cranston`" is a person, or that he is *particularly* known for portraying "`Walter White`"; the number of facts extracted depends on many factors, such as the domain, the ontology/KB used, the techniques employed, etc.

Listing 3: Relation Extraction and Linking example

```
Input:  Bryan Lee Cranston is an American actor. He
     ↪  is known for portraying ''Walter White'' in
     ↪  the drama series Breaking Bad.
Output:
dbr:Bryan_Cranston dbo:occupation dbr:Actor ;
   dbo:birthPlace dbr:United_States .
dbr:Walter_White_(Breaking_Bad) dbo:portrayer dbr:
     ↪ Bryan_Cranston ;
   dbo:series dbr:Breaking_Bad .
dbr:Breaking_Bad dbo:genre dbr:Drama .
```

Note that the previous listing provides direct binary relations. Many RE systems rather extract *n*-ary relations with generic role-based connectives. In Listing 4, we provide a real-world example given by the online FRED demo;[59] for brevity, we exclude some output triples not directly pertinent to the example.

Listing 4: FRED Relation Extraction example

```
Input:  Bryan Lee Cranston is an American actor. He
     ↪  is known for portraying "Walter White" in
     ↪ the drama series Breaking Bad.
Output (sample):
fred:Bryan_lee_cranston a fred:AmericanActor ;
   dul:hasQuality fred:Male .
fred:AmericanActor rdfs:subClassOf fred:Actor ;
   dul:hasQuality fred:American .
fred:know_1 a fred:Know ;
   vn.role:Theme fred:Bryan_lee_cranston ;
   fred:for fred:thing_1 .
fred:portray_1 a fred:Portray
  vn.role:Agent fred:thing_1 ;
  dul:associatedWith fred:walter_white_1 ;
  fred:in fred:Breaking_Bad .
fred:Breaking_Bad a fred:DramaSeries .
fred:DramaSeries dul:associatedWith fred:Drama ;
  rdfs:subClassOf fred:Series .
```

Here we see that relations are rather represented in an *n*-ary format, where for example, the relation "`portrays`" is represented as an RDF resource connected to the relevant entities by role-based predicates, where "`Walter White`" is given by the predicate `dul:associatedWith`, "`Breaking Bad`" is given by the predicate `fred:in`, and "`Bryan Lee Cranston`" is given by an indirect path of three predicates `vn.role:Agent/fred:for`$^{-}$`/vn.role:Theme`; the type of relation is then given as `fred:Portray`.

Applications: Of course, one of the main applications for the extraction, linking and representation of relations is to (further) populate Semantic Web knowledge-bases and ontologies; as a general application area, this has a variety of more (domain-)specific applications, where relation extraction has been applied for population in the domains of Medicine [254, 223], Terrorism [137], Sports [245], among others. An important application is for applying a structured form of *Discourse Representation*, where arguments implicit in a text are parsed, structured, and potentially linked with an ontology or KB to improve machine readability [98,9,102]. Another key application is that of *Question Answering* (Q&A) [279], whose purpose is to answer natural language questions over KBs, where approaches often begin by applying RE on the raw question text to gain an initial structure [309,296]. Other interesting applications have been to mine deductive inference rules from text [166], or for pattern recognition over text [111], or to verify or provide textual references for existing KB triples [96].

Process: The Relation Extraction and Linking (REL) process can vary depending on the particular methodology adopted. Some systems rely on traditional RE processes, where extracted relations are linked to a KB after extraction; other REL systems – such as those based on *distant supervision* – use binary relations in the KB to identify and generalize patterns from text mentioning the entities involved, which are then used to subsequently extract and link further relations. Generalizing, we structure this section as follows. First, many (mostly distant supervision) REL approaches begin by identifying named entities in the text, either through NER (generating raw mentions) or through EEL (additionally providing KB identifiers). Second, REL requires a method for parsing relations from text, which in some cases may involve using a traditional RE approach. Third, distant supervision REL approaches use existing KB relations to find and learn from example relation mentions, from which general patterns and/or features are extracted and used to generate novel relations. Fourth, an REL approach may apply a clustering procedure to group relations based on hypernymy or equivalence. Fifth, REL approaches – particularly those focused on extracting *n*-ary relations – must define an appropriate RDF representation to serialize output relations. Finally, in order to link the resulting relations to a given KB/ontology, REL often considers a mapping step to align identifiers.

System overview: Before discussing REL approaches in depth, we provide a general overview of the addressed systems, their purpose, publication year, and main techniques. As mentioned in the other sections, we only cover approaches that are directly related with the Semantic Web and that offer a peer-reviewed publication with implementation details. Although some

[59] `http://wit.istc.cnr.it/stlab-tools/fred/demo`

of the techniques addressed by the discussed papers can be extended to serve several domains, we rather only include details for scenarios explicitly evaluated in each study. Finally, the same naming convention is used as before: we use the system name if available, otherwise the surname of the first author and initials.

Artequakt (2003) [3] extracts relations from plain text using a domain ontology and linguistic resources. The approach uses syntactic analysis to identify verb-guided structures, an NER process, and text enrichment (WordNet) to later associate relation elements with the defined ontology. The output information is represented as XML and used to populate the given ontology.

***MintzBSJ* (2009) [200]** popularized the *distant supervision* (DS) paradigm for REL, matching entities in known Freebase relations to text using EEL, extracting features from sentences mentioning both entities in a Freebase relation under the assumption that those sentences mention the Freebase relation. These features are then applied to other entity-mention pairs to link their relation.

***RiedelYM* (2010) [242]** initially proposed a *multi-instance* learning model for applying DS-based REL, observing that many sentences mentioning an entity pair do not reflect the KB relation(s) between those pairs. They instead assume that each KB relation has *at least one* text mention and apply a collective model that pairs KB relations with specific mentions under that assumption.

MultiR (2011) [130] implements a DS-based method that adopts a multi-instance approach but focuses on the problem of overlapping relations whereby a relation mention is assumed to potentially refer to multiple KB relations; thus, this DS-based approach was the first to consider that a relation mention may refer to multiple KB relations (i.e., that mentions may have multiple valid *labels*).

***NguyenM* (2011) [219]** again apply a DS strategy for extracting YAGO triples from the text of Wikipedia articles, where kernel methods are applied over dependency and constituency parse trees extracted from the sentences (in Wikipedia) containing related pairs of YAGO entities.

PROSPERA (2011) [208] (*PRospering knOwledge with Scalability, PrEcision, and RecAll*) is a distributed framework for extracting relations based on patterns expressed as POS-tagged *n*-grams between two entities. To learn patterns, EEL is used to identify YAGO2 entities and thereafter relations in text. Constraint-based reasoning is used to filter negative/inconsistent relations.

BOA (2012) [107] (*BOotrsrapping the web of datA*) extract natural language patterns from text representing relations, where entities are linked to Semantic Web KBs and known KB relations are used to extract patterns from sentences containing two or more entities. Candidate patterns are filtered by various features in a machine learning process; top patterns are used for relation extraction according to the originating KB property.

DeepDive (2012) [220] uses a DS-based approach to extract Freebase triples from webpages using a distributed platform. The system extracts sentences encoding pairs of Freebase entities in known relations, generating a variety of features from dependency parse trees over which logistic regression classifiers are trained.

***ExnerN* (2012) [87]** extract triples from text by first applying Semantic Role Labeling (SRL), Coreference Resolution (CR) and EEL with respect to DBpedia. Existing DBpedia triples are matched to these output relations to discover general mapping rules that can be applied to other relations, generating new DBpedia triples.

Graphia (2012) [98] extracts complex relational dependencies from Wikipedia text using a novel representation called Structured Discourse Graphs (SDGs). These SDGs are computed by first applying syntactic parsing, then running CR and EEL over DBpedia, before transforming the resulting annotated parse tree to a graph representation.

LODifier (2012) [9] extracts entities and relationships to build an RDF representation linked to DBpedia and WordNet. They first apply EEL (Wikifier) and then use *Discourse Representation Theory* (DRT; using Boxer [65]) to obtain raw relations. Properties are then mapped to RDF WordNet. Finally, an RDF representation of the extracted relations is defined for the output.

MIML-RE (2012) [272] (*Multi-Instance Multi-Label RE*) performs DS-based REL, focusing on the labeling problem. Specifically, the approach considers that each pair of entities may appear in several relation mentions (multi-instance) and may refer to one of several KB relations between those entities (multi-label), devising an inference model for learning under such assumptions.

***TakamatsuSN* (2012) [273]** focus on the problem of incorrect labeling in DS-based approaches for REL. Noting that a relation mention with two KB entities may not reflect the KB relation(s), they propose a method to reduce incorrect labels using a generative model to predict whether or not the mention actually reflects a relation, thus refining labels before applying learning.

***LiuHLZLZ* (2013) [169]** propose a mapping of relations obtained by PATTY [209] to properties from YAGO2 by first obtaining property candidates (matching entity mentions from the relation) and then filtering them using various semantic similarity measures between the relation and each property candidate selected from the KB.

***Nebhi* (2013) [213]** extract DBpedia facts from text where entities are first extracted and linked to DBpedia; next binary relations are extracted using a syntactic parser. The extracted relations are then linked to existing DBpedia relations. For evaluation, they apply their method to a manually-labeled gold standard news corpus.

PATTY (2013) [209] extracts relations using a reference KB for typing purposes (YAGO2 or Freebase). Relation patterns are extracted from a dependency parse tree, annotated with KB types, and subsequently generalized to abstract patterns; these patterns are then grouped per synonymy or hypernymy to form a pattern taxonomy.

Wsabie (2013) [292] applies a DS-based REL approach based on embeddings that combine textual relation mentions with KB knowledge. More specifically, word- and relation-embeddings are computed to first match mentions to relations, while embeddings are also computed from the KB for entities and their properties in order to validate candidate KB relations.

RdfLiveNews (2013) [106] extracts RDF triples from the text of streaming RSS news feeds. After a text deduplication step, POS-tagging and NER are applied to derive an initial set of patterns, which are then filtered and refined according to an EEL step linking entities to existing relations in DBpedia. Patterns are then clustered using WordNet. Extracted relations are represented in RDF, where relations can be linked to existing KB properties.

Knowledge Vault (2014) [78] extracts Freebase triples from webpages at large scale. The process first applies EEL with respect to Freebase and then applies dependency parsing. Thereafter, existing triples in Freebase are used, in combination with distant supervision, to find patterns in sentences that indicate such relations. Logistic regression is used to classify triples taking positive/negative examples from the KB.

Refractive (2014) [88] focuses on scalability, proposing a distributed framework based on the Hadoop platform for extracting frame-based relations from English Wikipedia. First, dependency parsing is performed to obtain semantic *n*-ary relations (frames) that are later projected into binary relationships. The result can be stored with a Lucene index or serialized with RDF reification.

***DuttaMS* (2015) [82,83]** map facts obtained from the Web by OpenIE systems (NELL [201] and ReVerb [89]) to DBpedia instances and object properties. The main premise is to first map existing facts in DBpedia to relations produced by OpenIE systems, where the correspondences found are then clustered, filtered and extended to further OpenIE relations to create new DBpedia triples.

***AugensteinMC* (2016) [8]** propose another DS-based REL approach; their main focus is on incorporating CR with NER techniques to capture relations across sentences, as well as a custom seed-selection method that only uses relation mentions for training that do not involve entities with ambiguous labels in the KB to reduce noise.

FRED (2016) [102] extracts *n*-ary relations based on Discourse Representation Structures (DRSs) extracted by the Boxer tool [65], where labels are then assigned to these “boxes” using VerbNet [153] and FrameNet [12]. The boxes are then mapped to an RDF representation with RDFS/OWL axioms (see Listing 4 for an example).

***LinSLLS* (2016) [169]** propose a DS-based REL approach that leverages word embeddings trained using convolutional neural networks. In particular, word (and position) embeddings are used to create a vector sequence for each sentence; thereafter, for a given pair of entities, the vectors for all of their sentences are combined using *selective attention* to minimize the effect of noisy vectors.

Sar-graphs (2016) [155] (*Graphs of Semantically annotated relations*) proposes a graph-based modeling of the dependency structure of sentences that can be used for relation extraction. Sar-graphs are extracted based on finding sentences (using Bing search) encoding existing binary relations in a KB (Freebase), with the shortest path or spanning tree of the parse tree encompassing both entities being extracted and post-processed.

Fact Extractor (2017) [96] extracts *n*-ary relations (about football) from Wikipedia text using a frames-based approach. A custom repository of frames is created by combining FrameNet [12], Kicktionary [257] and other crowdsourced frames. These frames are used to extract *n*-ary relations whose entities are linked with DBpedia.

As before, an overview of the highlighted REL systems is provided in Table 4 summarizing a variety of features that we will discuss presently.

4.1. Entity Extraction (and Linking)

The first step of REL often consists of identifying entity mentions in the text. Here we distinguish three strategies, where, in general, many works follow the EEL techniques previously discussed in Section 2, particularly those adopting the first two strategies.

- The first strategy is to employ an end-to-end EEL system – such as DBpedia Spotlight [187], Wikifier [87], etc. – to match entities in the raw text. The benefit of this strategy is that KB identifiers are directly identified for subsequent phases.
- The second strategy is to employ a traditional NER tool – often from Stanford CoreNLP [8, 220,102,200,209] – and then later potentially link the resulting mentions to a KB. This strategy potentially has the benefit of identifying mentions of emerging entities, allowing to extract relations about entities not already in the KB.
- The third strategy is to rather skip the NER/EL phrase and rather directly apply an off-the-shelf RE/OpenIE tool or an existing dependency-based parser (discussed later) over the raw text to extract relational structures; such structures then embed parsed entity mentions over which EEL can be applied (potentially using an existing EEL system such as DBpedia Spotlight [98] or TagMe [91]). This has the benefit of using established RE techniques and potentially capturing emerging entities; however, such a strategy does not leverage knowledge of existing relations in the KB for extracting relation mentions (since relations are extracted prior to accessing the KB).

In the context of REL, when extracting and linking entities, Coreference Resolution (CR) plays a very important role. While other EEL applications may not require capturing every coreference in a text – e.g., it may be sufficient to capture that the entity is mentioned at least one in a document for semantic search or annotation tasks – in the context of REL, not capturing coreferences will potentially lose many relations. Consider again Listing 3, where the second sentence begins "`He is known for ...`"; CR is necessary to understand that "`He`" refers to "`Bryan Lee Cranston`", to extract that he portrays "`Walter White`", and so forth. In Listing 3, the portrays relation is connected (indirectly) to the node identifying "`Bryan Lee Cranston`"; this is possible because FRED uses Stanford CoreNLP's CR methods. A number of other REL systems [87,106] likewise apply CR to improve recall of relations extracted.

4.2. Parsing Relations

The next phase of REL systems often involves parsing structured descriptions from relation mentions in the text. The complexity of such structures can vary widely depending on the nature of the relation mention, the particular theory by which the mention is parsed, the use of pronouns, and so forth. In particular, while some tools rather extract simple binary relations of the form $p(s,o)$ with a designated subject–predicate–object, others may apply a more abstract semantic representation of *n*-ary relations with various dependent terms playing various roles.

In terms of parsing more simple binary relations, as mentioned previously, a number of tools use existing OpenIE systems, which apply a recursive extraction of relations from webpages, where extracted relations are used to guide the process of extracting further relations. In this setting, for example, Dutta *et al.* [83] use NELL [201] and ReVerb [89], Liu *et al.* [169] use PATTY [209], while Soderland and Mandhani [267] use TextRunner [14] to extract relations; these relations will later be linked with an ontology or KB.

In terms of parsing potentially more complex *n*-ary relations, a variety of methods can be applied. A popular method is to begin with a dependency-based parse of the relation mention. For example, Grafia [98] uses a Stanford PCFG parser to extract dependencies in a relation mention, over which CR and EEL are subsequently applied. Likewise, other approaches using a dependency parser to extract an initial syntactic structure from relation mentions include DeepDive [220], PATTY [209], Refractive [88] and works by Mintz et al [200], Nguyen and Moschitti [219], etc.

Other works rather apply higher-level theories of language understanding to the problem of modeling relations. One such theory is that of *frame semantics* [93], which considers that people understand sentences by recalling familiar structures evoked by a particular word; a common example is that of the term

Table 4

Overview of Relation Extraction and Linking systems

Entity denotes the NER or EEL strategy used; **Parsing** denotes the method used to parse relation mentions (Cons.: Constituency Parsing, Dep.: Dependency Parsing, DRS: Discourse Representation Structures, Emb.: Embeddings); **PS** refers to the Property Selection method (PG: Property Generation, RM: Relation Mapping, DS: Distant Supervision); **Rep.** refers to the reification model used for representation (SR: Standard Reification, BR: Binary Relation); **KB** refers to the main knowledge-base used; '—' denotes no information found, not used or not applicable

System	**Year**	**Entity**	**Parsing**	**PS**	**Rep.**	**KB**	**Domain**
Artequakt [3]	2003	GATE	Patterns	RM	BR	Artists ontology	Artists
AugensteinMC [8]	2016	Stanford	Features	DS	BR	Freebase	Open
BOA [107]	2012	DBpedia Spotlight	Patterns, Features	DS	BR	DBpedia	News, Wikipedia
DeepDive [220]	2012	Stanford	Dep., Features	DS	BR	Freebase	Open
DuttaMS [82,83]	2015	Keyword	OpenIE	DS	BR	DBpedia	Open
ExnerN [87]	2012	Wikifier	Frames	DS	BR	DBpedia	Wikipedia
Fact Extractor [96]	2017	Wiki Machine	Frames	DS	*n*-ary	DBpedia	Football
FRED [102]	2016	Stanford, TagMe	DRS	PG / RM	*n*-ary	DBpedia / BabelNet	Open
Graphia [98]	2012	DBpedia Spotlight	Dep.	PG	SR	DBpedia	Wikipedia
Knowledge Vault [78]	2014	—	Features	DS	BR	Freebase	Open
LinSLLS [167]	2016	Stanford	Emb.	DS	BR	Freebase	News
LiuHLZLZ [169]	2013	Stanford	Dep., Features	DS	BR	YAGO	News
LODifier [9]	2012	Wikifier	DRS	RM	SR	WordNet	Open
MIML-RE [272]	2012	Stanford	Dep., Features	DS	BR	Freebase	News, Wikipedia
MintzBSJ [200]	2009	Stanford	Dep., Features	DS	BR	Freebase	Wikipedia
MultiR [130]	2011	Stanford	Dep., Features	DS	BR	Freebase	Wikipedia
Nebhi [213]	2013	GATE	Patterns, Dep.	DS	BR	DBpedia	News
NguyenM [219]	2011	—	Dep., Cons.	DS	BR	YAGO	Wikipedia
PATTY [209]	2013	Stanford	Dep., Patterns	RM	—	YAGO	Wikipedia
PROSPERA [208]	2011	Keyword	Patterns	RM	BR	YAGO	Open
RdfLiveNews [106]	2013	DBpedia SpotLight	Patterns	PG / DS	BR	DBpedia	News
Refractive [88]	2014	Stanford	Frames	DS	SR	—	Wikipedia
RiedelYM [242]	2010	Stanford	Dep., Features	DS	BR	Freebase	News
Sar-graphs [155]	2016	Dictionary	Dep.	DS	—	Freebase / BabelNet	Open
TakamatsuSN [273]	2012	Hyperlinks	Dep.	DS	BR	Freebase	Wikipedia
Wsabie [292]	2013	Stanford	Dep., Features, Emb.	DS	BR	Freebase	News

"revenge", which evokes a structure involving various components, including the *avenger*, the *retribution*, the *target of revenge*, the *original victim being avenged*, and the *original offense*. These structures are then formally encoded as frames, categorized by the word senses that evoke the frame, encapsulating the constituents as frame elements. Various collections of frames have then been defined – with FrameNet [12] being a prominent example – to help identify frames and annotate frame elements in text. These frames can then be used to parse n-ary relations, as used for example by Refractive [88] or Fact Extractor [96].

A related theory used to parse complex n-ary relations is that of Discourse Representation Theory (DRT) [144], which offers a more logic-based perspective for reasoning about language. In particular, DRT is based on the idea of Discourse Representation Structures (DRS), which offer a first-order-logic (FOL) *style* representation of the claims made in language, incorporating n-ary relations, and even allowing negation, disjunction, equalities, and implication. The core idea is to build up a formal encoding of the claims made in a discourse spanning multiple sentences where the equality operator, in particular, is used to model coreference across sentences. These FOL style formulae are contextualized as boxes that indicate conjunction.

Tools such as Boxer [26] then allow for extracting such DRS "boxes" following a *neo-Davidsonian* representation, which at its core involves describing events. Consider the example sentence "Barack Obama met Raul Castro in Cuba"; we could consider representing this as $\text{meet}(\text{BO},\text{RC},\text{CU})$ with BO denoting "Barack Obama", etc. Now consider "Barack Obama met with Raul Castro in 2016"; if we represent this as $\text{meet}(\text{BO},\text{RC},2016)$, we see that the meaning of the third argument 2016 conflicts with the role of CU as a location earlier even though both are prefixed by the preposition "in". Instead, we will create an existential to represent the meeting; considering "Barack Obama met briefly with Raul Castro in 2016 while in Cuba", we could write (e.g.):

$$\exists e : \text{meet}(e), \text{Agent}(e,\text{BO}), \text{CoAgent}(e,\text{RC}),$$
$$\text{briefly}(e), \text{Theme}(e,\text{CU}), \text{Time}(e,2016)$$

where e denotes the event being described, essentially decomposing the complex n-ary relation into a conjunction of unary and binary relations.[60] Note that expressions such as $\text{Agent}(e,\text{BO})$ are considered as *semantic roles*, contrasted with syntactic roles; if we consider "Barack Obama met with Raul Castro", then BO has the syntactic role of *subject* and RC the role of *object*, but if we swap – "Raul Castro met with Barack Obama" – while the syntactic roles swap, we see little difference in semantic meaning: both BO and RC play the semantic role of *(co-)agents* in the event. The roles played by members in an event denoted by a verb are then given by various syntactic databases, such as VerbNet [153][61] and PropBank [152]. The Boxer [26] tool then uses VerbNet to create DRS-style boxes encoding such neo-Davidsonian representations of events denoted by verbs. In turn, REL tools such as LODifier [9] and FRED [102] (see Listing 4) use Boxer to extract relations encoded in these DRS boxes.

4.3. Distant Supervision

There are a number of significant practical shortcomings of using resources such as FrameNet, VerbNet, and PropBank to extract relations. First, being manually-crafted, they are not necessarily complete for all possible relations and syntactic patterns that one might consider and, indeed, are often only available in English. Second, the parsing method involved may be quite costly to run over all sentences in a very large corpus. Third, the relations extracted are complex and may not conform to the typically binary relations in the KB; creating *a posteriori* mappings may be non-trivial.

An alternative data-driven method for extracting relations – based on *distant supervision*[62] – has thus become increasingly popular in recent years, with a seminal work by Mintz *et al.* [200] leading to a flurry of later refinements and extensions. The core hypothesis behind this method is that given two entities with a known relation in the KB, sentences in which both entities are mentioned in a text are likely to also mention the relation. Hence, given a KB predicate (e.g., dbo:genre), we can consider the set of known binary relations between pairs of entities from the KB (e.g, (dbr:Breaking_Bad,dbr:Drama), (dbr:X_Files,dbr:Science_Fiction), etc.) with that predicate and look for sentences that mention both entities, hypothesizing that the sentence offers an example of a mention of that relation (e.g., "in the drama series Breaking Bad", or "one of the

[60] One may note that this is analogous to the same process of representing n-ary relations in RDF [123].

[61] See https://verbs.colorado.edu/verb-index/vn/meet-36.3.php

[62] Also known as *weak supervision* [130].

most popular Sci-Fi shows was X-Files"). From such examples, patterns and features can be generalized to find fresh KB relations involving other entities in similar such mentions appearing in the text.

The first step for such distant supervision methods is to find sentences containing mentions of two entities that have a known binary relation in the KB. This step essentially relies on the EEL process described earlier and can draw on techniques from Section 2. Note that examples may be drawn from external documents, where, for example, Sar-graphs [155] proposes to use Bing's web search to find documents containing both entities in an effort to build a large collection of example mentions for known KB relations. In particular, being able to draw from more examples allows for increasing the precision and recall of the REL process by finding better quality examples for training [220].

Once a list of sentences containing pairs of entities is extracted, these sentences need to be analyzed to extract patterns and/or features that can be applied to other sentences. For example, as a set of *lexical features*, Mintz *et al.* [200] proposes to use the sequence of words between the two entities, to the left of the first entity and to the right of the second entity; a flag to denote which entity came first; and a set of POS tags. Other features proposed in the literature include matching the label of the KB property (e.g., dbo:birthPlace – "Birth Place") and the relation mention for the associated pair of entities (e.g., "was born in") [169]; the number of words between the two entity mentions [107]; the frequency of *n*-grams appearing in the text window surrounding both entities, where more frequent *n*-grams (e.g., "was born in") are indicative of general patterns rather than specific details for the relation of a particular pair of entities (e.g., "prematurely in a taxi") [209], etc.

Aside from such lexical features, systems often parse the example relations to extract syntactic dependencies between the entities. A common method, again proposed by Mintz *et al.* [200] in the context of supervision, is to consider *dependency paths*, which are (shortest) paths in the dependency parse tree between the two entities; they also propose to include *window nodes* – terms on either side of the dependency path – as a syntactic feature to capture more context. Both the lexical and syntactic features proposed by Mintz *et al.* were then reused in a variety of subsequent related works using distant supervision, including Knowledge Vault [78], DeepDive [220], and many more besides.

Once a set of features is extracted from the relation mentions for pairs of entities with a known KB relation, the next step is to generalize and apply those features for other sentences in the text. Mintz *et al.* [200] originally proposed to use a multi-class logistic regression classifier: for training, the approach extracts all features for a given entity pair (with a known KB relation) across all sentences in which that pair appears together, which are used to train the classifier for the original KB relation; for classification, all entities are identified by Stanford NER, and for each pair of entities appearing together in some sentence, the same features are extracted from all such sentences and passed to the classifier to predict a KB relation between them.

A variety of works followed up on and further refined this idea. For example, Riedel *et al.* [242] note that many sentences containing the entity pair will not express the KB relation and that a significant percentage of entity pairs will have multiple KB relations; hence combining features for all sentences containing the entity pair produces noise. To address this issue, they propose an inference model based on the assumption that, for a given KB relation between two entities, *at least one* sentence (rather than all) will constitute a true mention of the relation; this is realized by introducing a set of binary latent variables for each such sentence to predict whether or not that sentence expresses the relation. Subsequently, for the MultiR system, Hoffman *et al.* [130] proposed a model further taking into consideration that relation mentions may *overlap*, meaning that a given mention may simultaneously refer to multiple KB relations; this idea was later refined by Surdeanu *et al.* [272], who proposed a similar *multi-instance multi-label* (MIML-RE) model capturing the idea that a pair of entities may have multiple relations (labels) in the KB and may be associated with multiple relation mentions (instances) in the text.

Another complication arising in learning through distant supervision is that of negative examples, where Semantic Web KBs like Freebase, DBpedia, YAGO, are necessarily incomplete and thus should be interpreted under an Open World Assumption (OWA): just because a relation is not in a KB, it does not mean that it is not true. Likewise, for a relation mention involving a pair of entities, if that pair does not have a given relation in the KB, it should not be considered as a negative example for training. Hence, to generate useful negative examples for training, the approach by Surdeanu *et al.* [272], Knowledge Vault [78], the approach by Min *et al.* [198], etc., propose a heuristic called (in [78]) a *Local Closed World Assumption* (LCWA), which assumes that if a relation $p(s,o)$ exists in the KB, then any relation $p(s,o')$ not in the KB is a negative example; e.g., if born(BO, US) exists in the KB, then born(BO, X) should be considered a negative example assuming it is not in the KB. While obviously this is far from infallible – working well for "functional-esque" properties like capital but less well for often multi-valued properties like child – it has proven useful in practice [272,198,78]: even if it produces false negative examples, it will produce far fewer than considering any relation not in the KB as false, and the benefit of having true negative examples amortizes the cost of potentially producing false negatives.

A further complication in distant supervision is with respect to noise in automatically labeled relation mentions caused, for example, by incorrect EEL results where entity mentions are linked to an incorrect KB identifier. To tackle this issue, a number of DS-based approaches include a *seed selection* process to try to select high-quality examples and reduce noise in labels. Along these lines, for example, Augenstein *et al.* [8] propose to filter DS-labeled examples involving ambiguous entities; for example, the relation mention "`New York is a state in the U.S.`" may be discarded since "`New York`" could be mistakenly linked to the KB identifier for the city, which may lead to a noisy example for a KB property such as *has-city*.

Other approaches based on distant supervision rather propose to extract generalized patterns from relation mentions for known KB relations. Such systems include BOA [107] and PATTY [209], which extract sequences of tokens between entity pairs with a known KB relation, replacing the entity pairs with (typed) variables to create generalized patterns associated with that relation, extracting features used to filter low-quality patterns; an example pattern in the case of PATTY would be "<PERSON> `is the lead-singer of` <MUSICBAND>" as a pattern for `dbo:bandMember` where, e.g., MUSICBAND indicates the expected type of the entity replacing that variable.

We also highlight a more recent trend towards alternative distant supervision methods based on embeddings (e.g., [292,302,167]). Such approaches have the benefit of not relying on NLP-based parsing tools, but rather relying on distributional representations of words, entities and/or relations in a fixed-dimensional vector space that, rather than producing a discrete parse-tree structure, provides a semantic representation of text in a (continuous) numeric space. Approaches such as proposed by Lin *et al.* [167] go one step further: rather than computing embeddings only over the text, such approaches also compute embeddings for the structured KB, in particular, the KB entities and their associated properties; these KB embeddings can be combined with textual embeddings to compute, for example, similarity between relation mentions in the text and relations in the KB.

We remark that tens of other DS-based approaches have recently been published using Semantic Web KBs in the linguistic community, most often using Freebase as a reference KB, taking an evaluation text collection from the New York Times (originally compiled by Riedel *et al.* [242]). While strictly speaking such works would fall within the scope of this survey, upon inspection, many do not provide any novel use of the KB itself, but rather propose refinements to the machine learning methods used. Hence we consider further discussion of such approaches as veering away from the core scope of this survey, particularly given their number. Herein, rather than enumerating all works, we have instead captured the seminal works and themes in the area of distant supervision for REL; for further details on distant supervision for REL in a Semantic Web setting, we can instead refer the interested reader to the Ph.D. dissertation of Augenstein [7].

4.4. Relation clustering

Relation mentions extracted from the text may refer to the same KB relation using different terms, or may imply the existence of a KB relation through hypernymy/sub-property relations. For example, mentions of the form "`X is married to Y`", "`X is the spouse of Y`", etc., can be considered as referring to the same KB property (e.g., `dbo:spouse`), while a mention of the form "`X is the husband of Y`" can likewise be considered as referring to that KB property, though in an implied form through hypernymy. Some REL approaches thus apply an analysis of such semantic relations – typically synonymy or hypernymy – to cluster textual mentions, where external resources – such as WordNet, FrameNet, VerbNet, PropBank, etc., – are often used for such purposes. These clustering techniques can then be used to extend the set of mentions/patterns that map to a particular KB relation.

An early approach performing such clustering was Artequakt [3], which leverages WordNet knowledge – specifically synonyms and hypernyms – to detect which pairs of relations can be considered equivalent or more specific than one another. A more recent version of such an approach is proposed by Gerber *et al.* [106] in the context of their RdfLiveNews system, where they define a similarity measure between relation patterns composed of a string similarity measure and a WordNet-based similarity measure, as well as the domain(s) and range(s) of the target KB property associated with the pattern; thereafter, a graph-based clustering method is applied to group similar patterns, where within each group, a similarity-based voting mechanism is used to select a single pattern deemed to represent that group. A similar approach was employed by Liu *et al.* [169] for clustering mentions, combining a string similarity measure and a WordNet-based measure; however they note that WordNet is not suitable for capturing similarity between terms with different grammatical roles (e.g., "`spouse`", "`married`"), where they propose to combine WordNet with a distributional-style analysis of Wikipedia to improve the similarity measure. Such a technique is also employed by Dutta *et al.* [83] for clustering relation mentions using a Jaccard-based similarity measure for keywords and a WordNet-based similarity measure for synonyms; these measures are used to create a graph over which Markov clustering is run.

An alternative clustering approach for generalized relation patterns is to instead consider the sets of entity pairs that each such pattern considers. Soderland and Mandhani [267] propose a clustering of patterns based on such an idea: if one pattern captures a (near) sub-set of the entity pairs that another pattern captures, they consider the former pattern to be subsumed by the latter and consider the former pattern to infer relations pertaining to the latter. A similar approach is proposed by Nakashole *et al.* [209] in the context of their PATTY system, where subsumption of relation patterns is likewise computed based on the sets of entity pairs that they capture; to enable scalable computation, the authors propose an implementation based on the MapReduce framework. Another approach along these lines – proposed by Riedel *et al.* [243] – is to construct what the authors call a *universal schema*, which involves creating a matrix that maps pairs of entities to KB relations and relation patterns associated with them (be it from training or test data); over this matrix, various models are proposed to predict the probability that a given relation holds between a pair of entities given the other KB relations and patterns the pair has been (probabilistically) assigned in the matrix.

4.5. RDF Representation

In order to populate Semantic Web KBs, it is necessary for the REL process to represent output relations as RDF triples. In the case of those systems that produce binary relations, each such relation will typically be represented as an RDF triple unless additional annotations about the relation – such as its provenance – are also captured. In the case of systems that perform EEL and a DS-style approach, it is furthermore the case that new IRIs typically will not need to be minted since the EEL process provides subject/object IRIs while the DS labeling process provides the predicate IRI from the KB; this process has the benefit of also directly producing RDF triples under the native identifier scheme of the KB. However, for systems that produce *n*-ary relations – e.g., according to frames, DRT, etc. – in order to populate the KB, an RDF representation must be defined. Some systems go further and provide RDFS/OWL axioms that enrich the output with well-defined semantics for the terms used [102].

The first step towards generating an RDF representation is to mint new IRIs for the entities and relations, etc., extracted. The BOA [107] framework proposes to first apply entity linking using a DS-style approach (where predicate IRIs are already provided), where for emerging entities not found in the KB, IRIs are minted based on the mention text. The FRED system [102] likewise begins by minting IRIs to represent all of the elements, roles, etc., produced by the Boxer DRT-based parser, thus *skolemizing* the events: grounding the existential variables used to denote such events with a constant (more specifically, an IRI).

Next, an RDF representation must be applied to structure the relations into RDF graphs. In cases where binary relations are not simply represented as triples, existing mechanisms for RDF reification – namely RDF *n*-ary relations, RDF reification, singleton properties, named graphs, etc. (see [123,253] for examples and more detailed discussion) – can, in theory, be adopted. In general, however, most systems define bespoke representations (most similar to RDF *n*-ary relations). Amongst these, Freitas *et al.* [98] propose a bespoke RDF-based discourse representation format that they call "Structured Discourse Graphs" capturing the subject, predicate and object of the relation, as well as (general) reification and temporal annotations; LODifier [9] maps Boxer output to RDF by mapping unary relations to `rdf:type` triples and binary relations to triples with a custom predicate, using RDF reification to represent disjunction, negation, etc., present in the DRS output; FRED [102] applies an *n*-ary–relation-style representation of the DRS-based relations extracted by Boxer, likewise mapping unary relations to `rdf:type` triples and binary relations to triples with a custom predicate (see Listing 4); and so forth.

Rather than creating a bespoke RDF representation, other systems rather try to map or project extracted relations directly to the native identifier scheme and data model of the reference KB. Likewise, those systems that first create a bespoke RDF representation may apply an *a posteriori* mapping to the KB/ontology. Such methods for performing mappings are now discussed.

4.6. Relation mapping

While in a distant supervision approach, the patterns and features extracted from textual relation mentions are directly associated with a particular (typically binary) KB property, REL systems based on other extraction methods – such as parsing according to legacy OpenIE systems, or frames/DRS theory – are still left to align the extracted relations with a given KB.

A common approach – similar to distant supervision – is to map pairs of entities in the parsed relation mentions to the KB to identify what known relations correspond to a given relation pattern.[63] This process is more straightforward when the extracted relations are already in a binary format, as produced, for example, by OpenIE systems. Dutta *et al.* [83] apply such an approach to map the relations extracted by OpenIE systems – namely the Nell and ReVerb tools – to DBpedia properties: the entities in triples extracted from such OpenIE systems are mapped to DBpedia by an EEL process, where existing KB relations are fed into an association-rule mining process to generate candidate mappings for a given OpenIE predicate and pair of entity-types; these rules are then applied over clusters of OpenIE relations to generate fresh DBpedia triples.

In the case of systems that natively extract *n*-ary relations – e.g., those systems based on frames or DRS – the process of mapping such relations to a binary KB relation – sometimes known as *projection* of *n*-ary relations [155] – is considerably more complex. Rather than trying to project a binary relation from an *n*-ary relation, some approaches thus rather focus on mapping elements of *n*-ary relations to classes in the KB. Such an approach is adopted by Gerber *et al.* [106] for mapping elements of binary relations extracted via learned patterns to DBpedia entities and classes. The FRED system [102] likewise provides mappings of its DRS-based relations to various ontologies and KBs, including WordNet and DOLCE ontologies (using WSD) and the DBpedia KB (using EEL).

On the other hand, other systems do propose techniques for projecting binary relations from *n*-ary relations and linking them with KB properties; such a process must not only identify the pertinent KB property (or properties), but also the subject and object entities for the given *n*-ary relation; furthermore, for DRS-style relations, care must be taken since the statement may be negated or may be part of a disjunction. Along those lines, Exner and Nugues [87] initially proposed to generate triples from DRS relations by means of a combinatorial approach, filtering relations expressed with negation. In follow-up work on the Refractive system, Exner and Nugues [88] later propose a method to map *n*-ary relations extracted through PropBank to DBpedia properties: existing relations in the KB are matched to extracted PropBank roles such that more matches indicate a better property match; thereafter, the subject and object of the KB relation are generalized to their KB class (used to identify subject/object in extracted relations), and the relevant KB property is proposed as a candidate for other instances of the same role (without a KB relation) and pairs of entities matching the given types. Legalo [237] proposes a method for mapping FRED results to binary KB relations by concatenating the labels of nodes on paths in the FRED output between elements identified as (potential) subject/object pairs, where these concatenated path labels are then mapped to binary KB properties to project new RDF triples. Rouces *et al.* [253], on the other hand, propose a rule-based approach to project binary relations from FrameNet patterns, where *dereification rules* are constructed to map suitable frames to binary triples by mapping frame elements to subject and object positions, creating a new predicate from appropriate conjugate verbs, further filtering passive verb forms with no clear binary relation.

4.7. Evaluation

REL is a challenging task, where evaluation is likewise complicated by a number of fundamental factors. In general, human judgment is often required to assess the quality of the output of systems performing such a task, but such assessments can often be subjective. Creating a gold-standard can likewise be complicated, particularly for those systems producing *n*-ary relations, requiring an expert informed on the par-

[63] More specifically, we distinguish between distant supervision approaches that use KB entities and relations to extract relation mentions (as discussed previously), and the approaches here, which extract such mentions without reference to the KB and rather map to the KB in a subsequent step, using matches between existing KB relations and parsed mentions to propose candidate KB properties.

ticular theory by which such relations are extracted; likewise, in DS-related scenarios, the expert must label the data according to the available KB relations, which may be a tedious task requiring in-depth knowledge of the KB. Rather than creating a gold-standard, another approach is to apply *a posteriori* assessment of the output by human judges, i.e., run the process over unlabeled text, generate relations, and have the output validated by human judges; while this would appear more reasonable for systems based on frames or DRS – where creating a gold-standard for such complex relations would be arduous at best – there are still problems in assessing, for example, recall.[64] Rather than relying on costly manual annotations, some systems rather propose automated methods of evaluation based on the KB where, for example, parts of the KB are withheld and then experiments are conducted to see if the tool can reinstate the withheld facts or not; however, such approaches offer rather approximate evaluation since the KB is incomplete, the text may not even mention the withheld triples, and so forth.

In summary, then, approaches for evaluating REL are quite diverse and in many cases there are no standard criteria for assessing the adequacy of a particular evaluation method. Here we discuss some of the main themes for evaluation, broken down by datasets used, how evaluators are employed to judge the output, how automated evaluation can be conducted, and what are the typical metrics considered.

Datasets Most approaches consider REL applied to general-domain text collections, such as Wikipedia articles, Newspaper articles, or even web pages. However, to simplify evaluation, many approaches may restrict REL to consider a domain-specific subset of such text collections, a fixed subset of KB properties or classes, and so forth. For example, Fossati *et al.* [96] focus their REL efforts on the Wikipedia articles about Italian soccer players using a selection of relevant frames; Augenstein *et al.* [8] apply evaluation for relations pertaining to entities in seven Freebase classes for which relatively complete information is available, using the Google Search API to find relevant documents for each such entity; and so forth.

A number of standard evaluation datasets have, however, emerged, particularly for approaches based on distant supervision. A widely reused gold-standard, for example, was that initially proposed by Riedel *et al.* [242] for evaluating their system, where they select Freebase relations pertaining to people, businesses and locations (corresponding also to NER types) and then link them with New York Times articles, first using Stanford NER to find entities, then linking those entities to Freebase, and finally selecting the appropriate relation (if any) to label pairs of entities in the same sentence with; this dataset was later reused by a number of works [130,272,243]. Other such evaluation resources have since emerged. Google Research[65] provides five REL corpora, with relation mentions from Wikipedia linked with manual annotation to five Freebase properties indicating institutions, date of birth, place of birth, place of death, and education degree. Likewise, the Text Analysis Conference often hosts a Knowledge Base Population (TAC–KBP) track, where evaluation resources relating to the REL task can be found[66]; such resources have been used and further enhanced, for example, by Surdeanu *et al.* [272] for their evaluation (whose dataset was in turn used by other works, e.g., by Min *et al.* [198], DeepDive [220], etc.). Another such initiative is hosted at the Europen Semantic Web Conference, where the Open Knowledge Extraction challenge (ESWC–OKE) has on one occasion hosted materials relating to REL using RDFa annotations on web-pages as labeled data.[67]

Note that all prior evaluation datasets relate to binary relations of the form *subject–predicate–object*. Creating gold standards for *n*-ary relations is complicated by the heterogeneity of representations that can be employed in terms of frames, DRS or other theories used. To address this issue, Gangemi *et al.* [103] proposed the construction of RDF graphs by means of logical patterns known as *motifs* that are extracted by the FRED tool and thereafter manually corrected and curated by evaluators to follow best Semantic Web practices; the result is a text collection annotated by instances of such motifs that can be reused for evaluation of REL tools producing similar such relations.

[64] Likewise we informally argue that a human judge presented with results of a system is more likely to confirm that output and give it the benefit of subjectivity, especially when compared with the creation of a gold standard where there is more freedom in choice of relations and more ample opportunity for subjectivity to be expressed.

[65] `https://code.google.com/archive/p/relation-extraction-corpus/downloads`

[66] For example, see `https://tac.nist.gov/2017/KBP/ColdStart/index.html`.

[67] For example, see Task 3: `https://2016.eswc-conferences.org/eswc-16-open-knowledge-extraction-oke-challenge`.

Evaluators In scenarios for which a gold standard is not available – or not feasible to create – the results of the REL process are often directly evaluated by humans. Many papers assign experts to evaluate the results, typically (we assume) authors of the papers, though often little detail on the exact evaluation process is given, aside from a rater agreement expressed as Cohen's or Fleiss' κ-measure for a fixed number of evaluators. Moving away from expert evaluation, some works have looked to leverage crowdsourcing platforms for labeling training and test datasets, where a broad range of users contribute judgments for a relatively low price. Amongst such works, we can mention Mintz *et al.* [200] using Amazon's Mechanical Turk[68] for evaluating relations, while Legalo [237] and Fossati *et al.* [96] use the Crowdflower[69] platform.

Automated evaluation Some works have proposed methods for performing automated evaluation of REL processes, in particular for testing DS-based methods. A common approach is to perform *held-out* experiments, where KB relations are (typically randomly) omitted from the training/DS phase and then metrics are defined to see how many KB relations are returned by the process, giving an indicator of recall; the intuition of such approaches is that REL is often used for completing an incomplete KB, and thus by holding back KB triples, one can test the process to see how many such triples the process can reinstate. Such an approach avoids expensive manual labeling but is not very suitable for precision since the KB is incomplete, and likewise assumes that held-out KB relations are both correct and mentioned in the text. On the other hand, such experiments can help gain insights at larger scales for a more diverse range of properties, and can be used to assess a relative notion of precision (e.g., to tune parameters), and have thus been used by Mintz *et al.* [200], Takamatsu *et al.* [273], Knowledge Vault [78], Lin *et al.* [167], etc. On the other hand, as mentioned previously, some works – including Knowledge Vault [78] – adopt a *partial Closed World Assumption* as a heuristic to generate negative examples taking into account the incompleteness of the KB; more specifically, extracted triples of the form (s,p,o') are labeled incorrect if (and only if) a triple (s,p,o) is present in the KB but (s,p,o') is not.

Metrics Standard evaluation measures are typically applied, including precision, recall, F-measure, accuracy, Area-Under-Curve (AUC–ROC), and so forth. However, given that relations may be extracted for multiple properties, sometimes macro-measures such as Mean Average Precision (MAP) are applied to summarize precision across all such properties rather than taking a micro precision measure [200,273,82]. Given the subjectivity inherent in evaluating REL, Fossati *et al.* [96] use a *strict* and *lenient* version of precision/recall/F-measure, where the former requires the relation to be exact and complete, while the latter also considers relations that are partially correct; relating to the same theme, the Legalo system includes *confidence* as a measure indicating the level of agreement and trust in crowdsourced evaluators for a given experiment. Some systems produce confidence or supports for relations, where P@k measures are sometimes used to measure the precision for the top-k results [200,242,169,167]. Finally, given that REL is inherently composed of several phases, some works present metrics for various parts of the task; as an example, for extracted triples, Dutta [83] considers a property precision (is the mapped property correct?), instance precision (are the mapped subjects/objects correct?), triple precision (is the extracted triple correct?), amongst other measures to indicate the ratio of triples extracted that are new to the KB, and so forth.

Third-party comparisons While a variety of REL papers include prior state-of-the-art approaches in their evaluations for comparison purposes, we not aware of a third-party study providing evaluation results of REL systems. Although Gangemi [101] provides a comparative evaluation of Alchemy, CiceroLite, FRED and ReVerb – all with public APIs available – for extracting relations from a paragraph of text on the Syrian war, he does not publish results for a linking phase; FRED is the only REL tool tested that outputs RDF.

4.8. Summary

This section presented the task of Relation Extraction and Linking in the context of the Semantic Web. The applications for such a task include KB Population, Structured Discourse Representation, Machine Reading, Question Answering, Fact Verification, amongst a variety of others. We discussed relevant papers following a high-level process consisting of: entity extraction (and coreference resolution), relation parsing, distant supervision, relation clustering, RDF

[68] https://www.mturk.com/mturk/welcome

[69] https://www.crowdflower.com/

representation, relation mapping, and evaluation. It is worth noting, however, that not all systems follow these steps in the presented order and not all systems apply (or even require) all such steps. For example, entity extraction may be conducted during relation parsing (where particular arguments can be considered as extracted entities), distant supervision does not require a formal representation nor relation-mapping phase, and so forth. Hence the presented flow of techniques should be considered illustrative, not prescriptive.

In general, we can distinguish two types of REL systems: those that produce binary relations, and those that produce *n*-ary relations (although binary relations can subsequently be projected from the latter tools [253,237]). With respect to binary relations, distant supervision has become a dominating theme in recent approaches, where KB relations are used, in combination with EEL and often CR, to find example mentions of binary KB relations, generalizing patterns and features that can be used to extract further mentions and, ultimately, novel KB triples; such approaches are enabled by the existence of modern Semantic Web KBs with rich factual information about a broad range of entities of general interest. Other approaches for extracting binary relations rather rely on mapping the results of existing OpenIE systems to KBs/ontologies. With respect to extracting *n*-ary relations, such approaches rely on more traditional linguistic techniques and resources to extract structures according to frame semantics or Discourse Representation Theory; the challenge thereafter is to represent the results as RDF and, in particular, to map the results to an existing KB, ontology, or collection thereof.

Choosing an RE strategy for a particular application scenario can be complex given that every approach has pros and cons regarding the application at hand. However, we can identify some key considerations that should be taken into account:

- *Binary vs. n-ary*: Does the application require binary relations or does it require *n*-ary relations? Oftentimes the results of systems that produce binary relations can be easier to integrate with existing KBs already composed of such, where DS-based approaches, in particular, will produce triples using the identifier scheme of the KB itself. On the other hand, *n*-ary relations may capture more nuances in the discourse implicit in the text, for example, capturing semantic roles, negation, disjunction, etc. in complex relations.
- *Identifier creation*: Does the application require finding and identifying new instances in the text not present in the KB? Does it require finding and identifying emerging relations? Most DS-based approaches do not consider minting new identifiers but rather focus on extracting new triples within the KB's universe (the sets of identifiers it provides). However, there are some exceptions, such as BOA [107]. On the other hand, most REL systems dealing with *n*-ary relations mint new IRIs as part of their output representation.
- *Language*: Does the application require extraction for a language other than English? Though not discussed previously, we note that almost all systems presented here are evaluated only for English corpora, the exceptions being BOA [107], which is tested for both English and German text; and the work by Fossati *et al.* [96], which is tested for Italian text. Thus in scenarios involving other languages, it is important to consider to what extent an approach relies on a language-specific technique, such as POS-tagging, dependency parsing, etc. Unfortunately, given the complexity of REL, most works are heavily reliant on such language-specific components. Possible solutions include trying to replace the particular component with its equivalent in another language (which has no guarantees to work as well as those tested in evaluation), or, as proposed for the FRED [102] tool, use an existing API (e.g., Bing!, Google, etc.) to translate the text to the supported language (typically English), with the obvious caveat of the potential for translation errors (though such services are continuously improving in parallel with, e.g., Deep Learning).
- *Scale & Efficiency*: In applications dealing with large text collections, scalability and efficiency become crucial considerations. With some exceptions, most of the approaches do not explicitly tackle the question of scale and efficiency. On the other hand, REL should be highly parallelizable given that processing of different sentences, paragraphs and/or documents can be performed independently assuming some globally-accessible knowledge from the KB; parallelization has been used, e.g., by Nakashole *et al.* [209], who cluster relational patterns using a distributed MapReduce framework. Indeed, initiatives such as Knowledge Vault – using standard DS-based REL techniques to extract 1.6 billion triples from a large-scale Web corpus – provide a practical demonstration that, with careful engineering and selection of

techniques, REL can be applied to text collections at a very large (potentially Web) scale.

- *Various other considerations*, such as availability or licensing of software, provision of an API, etc., may also need to be taken into account.

In summary, REL is a challenging task in Information Retrieval, where no single system technique or system can be considered to cover all use-cases. Indeed, REL is still very much an active area of research: from this survey, we can observe that REL is experiencing a surge in interest, where, in particular, the availability of Semantic Web KBs has lead to a variety of novel developments on distant supervision approaches for extracting binary relations.

5. Semi-structured Information Extraction

The primary focus of the survey – and the sections thus far – is on Information Extraction over unstructured text. However, the Web is full of semi-structured content, where HTML in particular allows for demarcating titles, links, lists, tables, etc., imposing a limited structure on documents. While it is possible to simply extract the text from such sources and apply previous methods, the structure available in the source, though limited, can offer useful hints for the IE process.

Hence a number of works have emerged proposing Information Extraction methods using Semantic Web languages/resources targeted at semi-structured sources. Some works are aimed at building or otherwise enhancing Semantic Web KBs (where, in fact, many of the KBs discussed originated from such a process [159,128]). Other works rather focus on enhancing or annotating the structure of the input corpus using a Semantic Web KB as reference. Some works make significant reuse of previously discussed techniques for plain text – particularly entity linking and sometimes relation extraction – adapted for a particular type of input document structure. Other works rather focus on custom techniques for extracting information from the particular structure of a particular data source.

Our goal in this section is thus to provide an overview of some of the most popular techniques and tools that have emerged in recent years for Information Extraction over semi-structured sources of data using Semantic Web languages/resources. Given that the techniques vary widely in terms of the type of structure considered, we organize this section differently from those that came before. In particular, we proceed by discussing two types of semi-structured sources – markup documents and tables – and discuss works that have been proposed for extracting information from such sources using Semantic Web KBs.

We do not include languages or approaches for mapping from one explicit structure to another (e.g., R2RML [68]), nor that rely on manual scraping (e.g., Piggy Bank [136]), nor tools that simply apply existing IE frameworks (e.g., Magpie [84], RDFaCE [147], SCMS [216]). Rather we focus on systems that extract and/or disambiguate entities, concepts, and/or relations from the input sources and that have methods adapted to exploit the partial structure of those sources (i.e., they do not simply extract and apply IE processes over flat text). Again, we only include proposals that in some way directly involve a Semantic Web standard (RDF(S)/OWL/SPARQL, etc.), or a resource described in those standards, be it to populate a Semantic Web KB, or to link results with such a KB.

5.1. Markup Documents

The content of the Web has traditionally been structured according to the HyperText Markup Language, which lays out a document structure for webpages to follow. While this structure is primarily perceived as a way to format, display and offer navigational links between webpages, it can also be – and has been – leveraged in the context of Information Extraction. Such structure includes, for example, the presence of hyperlinks, title tags, paths in the HTML parse tree, etc. Other Web content – such as Wikis – may be formatted in markup other than HTML, where we include frameworks for such formats here. We provide an overview of these works in Table 5. Given that all such approaches implement diverse methods that depend on the markup structure leveraged, we will not discuss techniques in detail. However, we will provide more detailed discussion for IE techniques that have been proposed for HTML tables in a following section.

COHSE (2008) [16] (*Conceptual Open Hypermedia Service*) uses a reference taxonomy to provide personalized semantic annotation and hyperlink recommendation for the current webpage that a user is browsing. A use-case is discussed for such annotation/recommendation in the biomedical domain, where a SKOS taxonomy can be used to recommend links to further material on more/-less specific concepts appearing in the text, with different types of users (e.g., doctors, the public) receiving different forms of recommended links.

Table 5

Overview of Information Extraction systems for Markup Documents

Task denotes the IE task(s) considered (EEL: Entity Extraction & Linking, CEL: Concept Extraction & Linking, REL: Relation Extraction & Linking); **Structure** denotes the type of document structure leveraged for the IE task;
'—' denotes no information found, not used or not applicable

System	**Year**	**Task**	**Source**	**Domain**	**Structure**	**KB**
COHSE [16]	2008	CEL	Webpages	Medical	Hyperlinks	Any (SKOS)
DBpedia [159]	2007	EEL/CEL/REL	Wikipedia	Open	Wiki	—
DeVirgilio [288]	2011	EEL/CEL	Webpages	Commerce	HTML (DOM)	DBpedia
Epiphany [2]	2011	EEL/CEL/REL	Webpages	Open	HTML (DOM)	Any (SPARQL)
Knowledge Vault [78]	2014	EEL/CEL/REL	Webpages	Open	HTML (DOM)	Freebase
Legalo [237]	2014	REL	Webpages	Open	Hyperlinks	—
LIEGE [259]	2012	EEL	Webpages	Open	Lists	YAGO
LODIE [105]	2014	EEL/REL	Webpages	Open	HTML (DOM)	Any (SPARQL)
RathoreR [264]	2014	CEL	Wikipedia	Physics	Titles	Custom ontology
YAGO (2007) [128]	2007	EEL/CEL/REL	Wikipedia	Open	Wiki	—

DBpedia (2007) [159] is a prominent initiative to extract a rich RDF KB from Wikipedia. The main source of extracted information comes from the semi-structured info-boxes embedded in the top right of Wikipedia articles; however, further information is also extracted from abstracts, hyperlinks, categories, and so forth. While much of the extracted information is based on manually-specified mappings for common attributes, components are provided for higher-recall but lower-precision automatic extraction of info-box information, including recognition of datatypes, etc.

***DeVirgilio* (2011) [288]** uses keyphrase extraction to semantically annotate web-pages, linking keywords to DBpedia. The approach breaks webpages down into "semantic blocks" describing specific elements based on HTML elements; keyphrase extraction is the applied over individual blocks. Evaluation is conducted in the E-Commerce domain, adding RDFa annotations using the Goodrelations vocabulary [121].

Epiphany (2011) [2] aims to semantically annotate webpages with RDFa, incorporating embedded links to existing Linked Data KBs. The process is based on an input KB, where labels of instances, classes and properties are extracted. A custom IE pipeline is then defined to chunk text and match it with the reference labels, with disambiguation performed based on existing relations in the KB for resolved entities. Facts from the KB are then matched to the resolved instances and used to embed RDFa annotations in the webpage.

Knowledge Vault (2014) [78] was discussed before in the context of Relation Extraction & Linking over text. However, the system also includes a component for extracting features from the structure of HTML pages. More specifically, the system extracts the Document Object Model (DOM) from a webpage, which is essentially a hierarchical tree of HTML tags. For relations identified on the webpage using a DS approach, the path in the DOM tree between both entities (for which an existing KB relation exists) is extracted as a feature.

Legalo (2014) [237] applies relation extraction based on the hyperlinks of webpages that describe entities, with the intuition that the anchor text (or more generalized context) of the hyperlink will contain textual hints about the relation between both entities. More specifically, a frame-based representation of the textual context of the hyperlink is extracted and linked with a KB; next, to create a label for a direct binary relation (an RDF triple), rules are applied on the frame-based representation to concatenate labels on the shortest path, adding event and role tags. The label is then linked to properties in existing vocabularies.

LIEGE (2012) [259] (*Link the entIties in wEb lists with the knowledGe basE*) performs EEL with respect to YAGO and Wikipedia over the text elements of HTML lists embedded in webpages. The authors propose specific features for disambiguation in the context of such lists where, in particular, the main assumption is that the entities appearing in a HTML list will often correspond to the same concept; this intuition is captured with a similarity-based measure that, for a given list, computes the distance of the types of candidate entities in the class hierarchy of YAGO. Other typical disambiguation features for EEl, such as prior probability, keyword-based similarities between entities, etc., are also applied.

LODIE (2014) [105] propose a method for using Linked Data to perform enhanced *wrapper induction*: leveraging the often regular structure of webpages on the same website to extract a mapping that serves to extract information in bulk from all its pages. LODIE then proposes to map webpages to an existing KB to identify the paths in the HTML parse tree that lead to known entities for concepts (e.g., movies), their attributes/relations (e.g., runtime, director), and associated values. These learned paths can then be applied to unannotated webpages on the site to extract further (analogous) information.

RathoreR (2014) [264] focus on topic extraction for webpages guided by a reference ontology. The overall process involves applying keyphrase extraction over the textual content of the webpage, mapping the keywords to an ontology, and then using the ontology to decide the topic. However, the authors propose to leverage the structure of HTML, where keywords extracted from the title, the meta-tags or the section-headers are analyzed first; if no topic is found, the process resorts to using keywords from the body of the document.

YAGO (2007) [128] is another major initiative for extracting information from Wikipedia in order to create a Semantic Web KB. Most information is extracted from info-boxes, but also from categories, titles, etc. The system also combines information from GeoNames, which provides geographic context; and WordNet, which allows for extracting cleaner taxonomies from Wikipedia categories. A distinguishing aspect of YAGO2 is the ability to capture temporal information as a first-class dimension of the KB, where entities and relations/attributes are associated with a hierarchy of properties denoting start/end dates.

It is interesting to note that KBs such as DBpedia [159] and YAGO2 [128] – used in so many of the previous IE works discussed throughout the survey – are themselves the result of IE processes, particularly over Wikipedia. This highlights something of a "snowball effect", where as IE methods improve, new KBs arise, and as new KBs arise, IE methods improve.[70]

5.2. *Tables*

Tabular data is common on the Web, where HTML tables embedded in webpages are plentiful and often contain rich, semi-structured, factual information [32, 61]. Hence, extracting information from such tables is indeed a tempting prospect. However, Web tables are primarily designed with human readability in mind rather than machine readability. Web tables, while numerous, can thus be highly heterogeneous and idiosyncratic: even tables describing similar content can vary widely in terms of structuring that content [61]. More specifically, the following complications arise when trying to extract information from such tables:

- Although Web tables are easy to identify (using the `<table>` HTML tag), many Web tables are used purely for layout or other presentational purposes (e.g., navigational sidebars, forms, etc.); thus, a preprocessing step is often required to isolate factual tables from HTML [32].
- Even tables containing factual data can vary greatly in structure: they may be "transposed", or may simply list attributes in one column and values in another, or may represent a matrix of values. Sometimes a further subset – called "relational tables" [32] – are thus extracted, where the table contains a column header, with subsequent rows comprising tuples in the relation.
- Even relational tables may contain irregular structure, including cells with multiple rows separated by an informal delimiter (e.g., a comma), nested tables as cell values, merged cells with vertical and/or horizontal orientation, tables split into various related sections, and so forth [231].
- Although column headers can be identified as such using (`<th>`) HTML tags, there is no fixed schema: for example, columns may not always have a fixed domain of values, there may be no obvious primary key or foreign keys, there may be hierarchical (i.e., multi-row) headers; etc.

[70] Though of course, we should not underestimate the value of Wikipedia itself as a raw source for IE tasks.

– Column names and cell values often lack clear identifiers or typing: Web tables often contain potentially ambiguous human-readable labels.

There have thus been numerous works on extracting information from tables, sometimes referred to as *table interpretation*, *table annotation*, etc. (e.g., [42,231, 32,61,286,297], to name some prominent works). The goal of such works is to interpret the implicit structure of tables so as to categorize them for search; or to integrate the information they contain and enable performing joins over them, be it to extend tables with information from other tables, or extracting the information to an external unified representation that can be queried.

More recently, a variety of approaches have emerged using Semantic Web KBs as references to help with extracting information from tables (sometimes referred to as *semantic table interpretation*, *semantic table annotation*, etc.). We discuss such approaches herein.

Process: While proposed approaches vary significantly, more generally, given a table and a KB, such works aim to link tables/columns to KB classes, link columns or tuples of columns to KB properties, and link individual cells to KB entities. The aim can then be to annotate the table with respect to the KB (useful for, e.g., later integrating or retrieving tables), and indeed to extract novel entities or relations from the table to further populate the KB. Hence we consider this an IE scenario. While methods discussed previously for IE over unstructured sources can be leveraged for tables, the presence of a tabular structure does suggest the applicability of novel features for the IE process. For example, one might expect in some tables to find that elements of the same column pertain to the same type, or pairs of entities on the same row to have a similar relation as analogous pairs on other rows. On the other hand, cells in a table have a different textual context, which may be the caption, the text referring to the table, etc., rather than the surrounding text; hence, for example, distributional approaches intended for text may not be *directly* applicable for tables.

Example: Consider a HTML table embedded in a webpage about the actor Bryan Cranston as follows:

Character	Series	Network	Ep.
Uncle Russell	Raising Miranda	CBS	9
Hal	Malcolm in the Middle	Fox	62
Walter	Breaking Bad	AMC	151
Lucifier Light Bringer	Fallen	ABC	4
Vince	Sneaky Pete	Amazon	10

We see that the table contains various entities, and that entities in the same column tend to correspond to a particular type. We also see that entities on each row often have implicit relations between them, organized by column; for example, on each row, there are binary relations between the elements of the `Character` and `Series` columns, the `Series` and `Networks` columns, and (more arguably in the case that multiple actors play the same character) between the `Character` and `Ep.` columns. Furthermore, we note that some relations exist from Bryan Cranston – the subject of the webpage – to the elements of various columns of the table.[71]

The approaches we enumerate here attempt to identify entities in table cells, assign types to columns, extract binary KB relations across columns, and so forth.

However, we also see some complications in the table structure, where some values span multiple cells. While this particular issue is relatively trivial to deal with – where simply duplicating values into each spanned cell is effective [231] – a real-world corpus of (HTML) tables will exhibit many further such complications; here we gave a relatively clean example.

Systems: We now discuss works that aim to extract entities, concepts or relations from tables, using Semantic Web KBs. We also provide an overview of these works in Table 6.[72]

AIDA (2011) [129] is primarily an Entity Linking tool (discussed in more detail previously in Section 2), but it provides parsers for extracting and linking entities in HTML tables; however, no table-specific features are discussed in the paper.

DRETa (2014) [205] aims to extract relations in the form of DBpedia triples from Wikipedia's tables. The process uses internal Wikipedia hyperlinks in tables to link cells to DBpedia entities. Relations are then analyzed on a row-by-row basis, where an existing relation in DBpedia between two entities in one row is postulated as a candidate relation for pairs of entities in the corresponding columns of other rows; implicit relations from the entity of the article containing the table and the

[71] In fact, we could consider each tuple as an n-ary relation involving Bryan Cranston; however, this goes more towards a Direct Mapping representation of the table [75,5]; rather the methods we discuss focus on extraction of binary relations.

[72] We also note that many such works were covered by the recent survey of Ristoski and Paulheim [246], but with more of an emphasis on data mining aspects. We are interested in such papers from a related IE perspective where raw entities/concepts/relations are extracted; hence they are also included here for completeness.

Table 6

Overview of Information Extraction systems for Tables

EL and **RE** denotes the Entity Linking and Relation Extraction strategies used; **Annotation** denotes the elements of the table annotated by the approach (P: Protagonist, E: Entities, S: Subject column, T: Column types, R: Relations, T′: Table type); **KB** denotes the reference KB used (WDC: WebDataCommons, BTC: Billion Triple Challenge 2014)
'—' denotes no information found, not used or not applicable

System	**Year**	**EL**	**RE**	**Annotation**	**KB**	**Domain**
AIDA [129]	2011	AIDA	—	E	YAGO	Wikipedia
DRETa [205]	2014	Wikilinks	Features	PER	DBpedia	Wikipedia
Knowledge Vault [78]	2014	—	Features	ER	Freebase	Web
LimayeSC [164]	2010	Keyword	Features	ETR	YAGO	Wikipedia, Web
MSJ Engine [160]	2015	—	—	EST	WDC, BTC	Web
MulwadFJ [204]	2013	Keyword	Features	ETR	DBpedia, YAGO	Wikipedia, Web
ONDINE [28]	2013	Keyword	Features	ETR	Custom ontology	Microbes, Chemistry, Aeronautics
RitzeB [247]	2017	Various	Features	ESRT′	Web	DBpedia
TabEL [19]	2015	String-based	—	E	Wikipedia	YAGO
TableMiner^{+} [305]	2017	Various	Features	ESTR	Freebase	Wikipedia, Movies, Music
ZwicklbauerEGS [311]	2015	—	—	T	DBpedia	Wikipedia

entities in each column of the table are also considered for generating candidate relations. These relations – extracted as DBpedia triples – are then filtered using classifiers that consider a range of features for the source cells, columns, rows, headers, etc., thus generating the final triples.

Knowledge Vault (2014) [78] describe the extraction of relations from 570 million Web tables. First, an EEL process is applied to identify entities in the table. Next, these entities are matched to Freebase and compared with existing relations. These relations are then proposed as candidates relations between the two columns of the table in question. Thereafter, ambiguous columns are discarded with respect to the existing KB relations and extracted facts are associated a confidence based on the EEL process. A total of 9.4 million Freebase facts are extracted in the final result.

***LimayeSC* (2010) [164]** propose a probabilistic model that, given YAGO as a reference KB and a Web table as input, simultaneously assigns entities to cells, types to columns, and relations to pairs of columns. The core intuition is that the assignment of a candidate to one of these three aspects affects the assignment of the other two, and hence a collective assignment can boost accuracy. A variety of features are thus defined over the table in relation to YAGO, over which joint inference is applied to optimize a collective assignment.

MSJ Engine (2015) [160] (*Mannheim Search Join Engine*) aims to extend a given input (HTML) table with additional attributes (columns) and associated values (cells) using a reference data corpus comprising of Linked Data KBs and other tables. The engine first identifies a "subject" column of the input table deemed to contain the names of the primary entities described; the datatype (domain) of other columns is then identified. This meta-description is used to search for other data with the same entities using information retrieval techniques. Thereafter, retrieved tables are (left-outer) joined with the input table based on a fuzzy match of columns, using the attribute names, ontological hierarchies and instance overlap measures.

***MulwadFJ* (2013) [204]** aim to annotate tables with respect to a reference KB by linking columns to classes, cells to (fresh) entities or literals, and pairs of columns to properties denoting their relation. The KB that they consider combines DBpedia, YAGO and Wikipedia. Candidate entities are derived using keyword search on the cell value and surrounding values for context; candidate column classes are taken as the union of all classes in the KB for candidate entities in that column; candidate relations for pairs of columns are chosen based on existing KB relations between candidate entities in those columns; thereafter, a joint inference step is applied to select a suitable collective assignment of cell-to-entity, column-to-class and column-pair-to-property mappings.

ONDINE (2013) [28] uses specialized ontologies to guide the annotation and subsequent extraction of information from Web tables. A core ontology encodes general concepts, unit concepts for quantities, and relations between concepts. On the other hand, a domain ontology is used to capture a class hierarchy in the domain of extraction, where classes are associated with labels. Table columns are then categorized by the ontology classes and tuples of columns are categorized by ontology relations, using a combination of cosine-similarity matching on the column names and the column values. Fuzzy sets are then used to represent a given annotation, encoding uncertainty, with an RDF-based representation used to represent the result. The extracted fuzzy information can then be queried using SPARQL.

RitzeB **(2017) [247]** enumerate and evaluate a variety of features that can be brought to bear for extracting information from tables. They consider a taxonomy of features that covers: features extracted from the table itself, including from a single (header/value) cell, or multiple cells; and features extracted from the surrounding context of the table, including page attributes (e.g., title) or free text. Using these features, they then consider three matching tasks with respect to DBpedia and an input table: row-to-entity, column-to-property, and table-to-class, where various linking strategies are defined. The scores of these matchers are then aggregated and tested against a gold standard to determine the usefulness of individual features, linking strategies and aggregation metrics on precision/recall of the resulting assignments.

TabEL (2015) [19] focuses on the task of EEL for tables with respect to YAGO, where they begin by applying a standard EEL process over cells: extracting mentions and generating candidate KB identifiers. Multiple entities can be extracted per cell. Thereafter, various features are assigned to candidates, including prior probabilities, string similarity measures, and so forth. However, they also include special features for tables, including a repetition feature to check if the mention has been linked elsewhere in the table and also a measure of semantic similarity for entities assigned to the same row or table; these features are encoded into a model over which joint inference is applied to generate a collective assignment.

TableMiner$^+$ (2017) [305] annotates tables with respect to Freebase by first identifying a subject column considered to contain the names of the entities being primarily described. Next, a learning phase is applied on each entity column (distinguished from columns containing datatype values) to annotate the column and the entities it contains; this process can involve sampling of values to increase efficiency. Next, an update/refinement phase is applied to collectively consider the (keyword-based) similarity across column annotations. Relations are then extracted from the subject column to other columns based on existing triples in the KB and keyword similarity metrics.

ZwicklbauerEGS **(2013) [311]** focus on the problem of assigning a DBpedia type to each column of an input table. The process involves three steps. First, a set of candidate identifiers is extracted for each cell. Next, the types (both classes and categories) are extracted from each candidate. Finally, for a given column, the type most frequently extracted for the entities in its cells is assigned as the type for that column.

Summary: Hence we see that exploring custom IE processes dedicating to tabular input formats using Semantic Web KBs is a burgeoning but still relatively recent area of research; techniques combine a mix of traditional IE methods as described previously, as well as novel low-level table-specific features and high-level global inference models that capture the dependencies in linking between different columns of the same table, different cells of the same column or row, etc.

Also, approaches vary in what they annotate. For example, while Zwicklbauer *et al.* [311] focus on typing columns, and AIDA [129] and TabEL [19] focus on annotating entities, most works annotate various aspects of the table, in particular for the purposes of extracting relations. Amongst those approaches extracting relations, we can identify an important distinction: those that begin by identifying a subject column to which all other relations extend [160,247,305], and those that rather extract relations between any pair of columns in the table [205,78,164,204,28]. All approaches that we found for relation extraction, however, rely on extracting a set of features and then applying machine learning methods to classify likely-correct relations; similarly, almost all approaches rely on a "distant supervision" style algorithm where seed relations in the KB appearing in rows of the table are used as a feature to identify candidate relations between column pairs. In terms of other annotations, we note that DRETa [205] extracts the protagonist of a table as the main entity about which the containing webpage is about (considered an entity with possible relations to entities in the table), while RitzeB [247] extract a type for each table based on the type(s) of entities in the subject column.

5.3. *Other formats*

Information Extraction has also been applied to various other formats in conjunction with Semantic Web KBs and/or ontologies. Amongst these, a number of works have proposed specialized EEL techniques for multimedia formats, including approaches for performing EEL with respect to images [15], audio (speech) [17,239], and video [290,191,163]. Other works have focused on IE techniques in the context of social platforms, such as for Twitter [299,300,72], tagging systems [269,148], or for other user-generated content, such as keyword search logs [58], etc.

Similar techniques to those described have also been applied to structured input formats, including Semantic Web KBs themselves. For example, a variety of approaches have been recently proposed to model topics for Semantic Web KBs themselves, either to identify the main topics within a KB, or to identify related KBs [24,226,265,251]. However, given that such methods apply to already well-structured input formats, these works veer away from pure Information Extraction and head more towards the related areas of Data Mining and Knowledge Discovery – as discussed already in a recent survey by Ristoski and Paulheim [246] – where the goal is to extract high-level patterns from data for applications including KB refinement, recommendation tasks, clustering, etc. We thus consider such works as outside the current scope.

6. Discussion

In this survey, we have discussed a wide variety of works that lie at the intersection of the Information Extraction and Semantic Web areas. In particular, we discussed works that extract entities, concepts and relations from unstructured and semi-structured sources, linking them with Semantic Web KBs/ontologies.

Trends: The works that we have surveyed span almost two decades. Interpreting some trends from Tables 1, 3, 4, 5 & 6, we see that earlier works (prior to *ca.* 2009) in this intersection related more specifically to Information Extraction tasks that were either intended to build or populate domain-specific ontologies, or were guided by such ontologies. Such ontologies were assumed to model the conceptual domain under analysis but typically without providing an extensive list of entities; as such, traditional IE methods were used involving NER of a limited range of types, machine-learning models trained over manually-labeled corpora, handcrafted linguistic patterns to bootstrap extraction, generic linguistic resources such as WordNet for modeling word sense/hypernyms/synsets, etc.

However, post 2009, we notice a shift towards using general-domain KBs – DBpedia, Freebase, YAGO, etc. – that provide extensive lists of entities (with labels and aliases), a wide variety of types and categories, graph-structured representations of cross-domain knowledge, etc. We also see a related trend towards more statistical, data-driven methods. We posit that this shift is due to two main factors: (i) the expansion of Wikipedia as a reference source for general domain knowledge – and related seminal works proposing its exploitation for IE tasks – which, in turn, naturally translate into using KBs such as DBpedia and YAGO extracted from Wikipedia; (ii) advancement in statistical NLP techniques that emphasize understanding of language through relatively shallow analyses of large corpora of text (for example, techniques based on the distributional hypothesis) rather than use of manually crafted patterns, training over labeled resources, or deep linguistic parsing. Of course, we also see works that blend both worlds, making the most of both linguistic and statistical techniques in order to augment IE processes.

Another general trend we have observed is one towards more "holistic" methods – such as collective assignment, joint models, etc. – that consider the interdependencies implicit in extracting increasingly rich machine-readable information from text. On the one hand, we can consider intra-task dependencies being modeled where, for example, linking one entity mention to a particular KB entity may affect how other surrounding entities are linked. On the other hand, in more recent works, more and more we can see inter-task dependencies being modelled, where the tasks of NER and EL [171,218], or WSD and EEL [203,134], or EEL and REL [8], etc., are seen as interdependent. We see this trend of jointly modeling several interrelated aspects of IE as set to continue, following the idea that improving IE methods requires looking at the "bigger picture" and not just one aspect in isolation.

Communities: In terms of the 106 highlighted papers in this survey for EEL, CEL, REL and Semi-Structured Inputs – i.e., those papers referenced in the first columns of Tables 1, 3, 4, 5 & 6 – we performed a meta-analysis of the venues (conferences or journals) at which they were published, and the primary area(s) associated with that venue. The results are compiled in Table 7, showing 18 (of 55) venues with at least two such papers; for compiling these results, we count workshops and satellite events under the conference with which they were co-located. While Semantic Web venues top the list, we notice a significant number of papers in venues associated with other areas.

In order to perform a higher-level analysis of the areas from which the highlighted works have emerged, we mapped venues to areas (as shown for the venues in Table 7). In some cases the mapping from venues to areas was quite clear (e.g., ISWC → Semantic Web), while in others we chose to assign two main areas to a venue (e.g., WSDM → Web / Data Mining). Furthermore, we assigned venues in multidisciplinary or otherwise broader areas (e.g., Information Science) to a general classification: *Other*. Table 8 then aggregates the areas in which all highlighted papers were published; in the case that a paper is published at a venue assigned to two areas, we count the paper as +0.5 in each area. The table is ordered by the total number of highlighted papers published. In this analysis, we see that while the plurality of papers comes from the Semantic Web community, the majority (roughly two-thirds) do not, with many coming from the NLP, AI and DB communities, amongst others. We can also see, for example, that NLP papers tend to focus on unstructured inputs, while Database and Data Mining papers rather tend to target semi-structured inputs.

Most generally, we see that works developing Information Extraction techniques in a Semantic Web context have been pursued within a variety of communities; in other words, the use of Semantic Web KBs has become popular in variety of other (non-SW) research communities interested in Information Extraction.

Final remarks: Our goal with this work was to provide not only a comprehensive survey of literature in the intersection of the Information Extraction and Semantic Web areas, but also to – insofar as possible – offer an introductory text to those new to the area.

Table 7

Top Venues for Highlight Papers

Venue denotes publication series, **Area(s)** denotes the primary CS area(s) of the venue; **E/C/R/S** denote counts of highlighted papers in this survey relating to Entities, Concepts, Relations and Semi-Structured input, resp.; Σ denotes the sum of **E** + **C** + **R** + **S**.

Venue	**Area(s)**	**E**	**C**	**R**	**S**	Σ
ISWC	SW	3	3	4	3	13
Sem. Web J.	SW			4	2	6
ACL	NLP			5		5
EKAW	SW	1	1	1	2	5
ESWC	SW	2	2	1		5
EMNLP	NLP	2		2		4
J. Web Sem.	SW	1	1	1	1	4
WWW	Web	3		1		4
Int. Sys.	AI		2	1		3
SIGKDD	DM			1	2	3
WSDM	DM/Web		1	1	1	3
AIRS	IR			2		2
CIKM	DB/IR	2				2
JASIST	*Other*		2			2
NLDB	NLP/DB		2			2
OnTheMove	DB/SW		1		1	2
PVLDB	DB				2	2
Trans. ACL	NLP	2				2

Table 8

Top Areas for Highlight Papers

E/C/R/S denote counts of highlighted papers in this survey relating to Entities, Concepts, Relations and Semi-Structured input, resp.; Σ denotes the sum of **E** + **C** + **R** + **S**.

Area	**E**	**C**	**R**	**S**	Σ
Semantic Web (SW)	8	11.5	11	8.5	39
Nat. Lang. Proc. (NLP)	5	3	8	1	17
Art. Intelligence (AI)	3	6	2	1	12
Databases (DB)	2	1.5	2.5	4	10
Other		7		2	9
Data Mining (DM)		0.5	1.5	4	6
Information Retr. (IR)	2	2	2		6
Web	3	0.5	1.5	0.5	5.5
Machine Learning (ML)		1	0.5		1.5
Total	23	33	29	21	106

Hence we have focused on providing a survey that is as self-contained as possible, including a primer on traditional IE methods, and thereafter an overview on the extraction and linking of entities, concepts and relations, both for unstructured sources (the focus of the survey), as well as an overview of such techniques for semi-structured sources. In general, methods for extracting and linking relations, for example, often rely on methods for extracting and linking entities, which in turn often rely on traditional IE and NLP techniques. Along similar lines, techniques for Information Extraction over semi-structured sources often rely heavily on similar techniques used for unstructured sources. Thus, aside from providing a literature survey for those familiar with such areas, we believe that this survey also offers a useful entry-point for the uninitiated reader, spanning all such interrelated topics.

Likewise, as previously discussed, the relevant literature has been published by various communities, using sometimes varying terminology and techniques, with different perspectives and motivation, but often with a common underlying (technical) goal. By drawing the literature from different communities together, we hope that this survey will help to bridge such communities and to offer a broader understanding of the research literature lying at this busy intersection where Information Extraction meets the Semantic Web.

Acknowledgements: *This work was funded in part by Millennium Nucleus Center for Semantic Web Research, Grant No. NC120004 and Fondecyt, Grant No. 11140900. We would also like to thank Henry Rosales-Méndez and Ana B. Rios-Alvarado for their comments on the survey.*

References

[1] Farhad Abedini, Fariborz Mahmoudi, and Amir Hossein Jadidinejad. From Text to Knowledge: Semantic Entity Extraction using YAGO Ontology. *International Journal of Machine Learning and Computing*, 1(2):1–7, 2011.

[2] Benjamin Adrian, Jörn Hees, Ivan Herman, Michael Sintek, and Andreas Dengel. Epiphany: Adaptable rdfa generation linking the web of documents to the web of data. In *Knowledge Engineering and Knowledge Management (EKAW)*, pages 178–192. Springer, 2010.

[3] Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[4] Mehdi Allahyari and Krys Kochut. Automatic topic labeling using ontology-based topic models. In *International Conference on Machine Learning and Applications (ICMLA)*, pages 259–264. IEEE, 2015.

[5] Marcelo Arenas, Alexandre Bertails, Eric Prud'hommeaux, and Juan Sequeda. A Direct Mapping of Relational Data to RDF. W3C Recommendation, September 2012.

[6] Sophie Aubin and Thierry Hamon. Improving Term Extraction with Terminological Resources. In *International Conference on NLP, FinTAL*, pages 380–387. Springer, 2006.

[7] Isabelle Augenstein. *Web Relation Extraction with Distant Supervision.* PhDthesis, The University of Sheffield, July 2016.

[8] Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Distantly supervised Web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, 2016.

[9] Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph. Lodifier: Generating linked data from unstructured text. In *Extended Semantic Web Conference (ESWC)*, pages 210–224. Springer, 2012.

[10] Akalya B. and Nirmala Sherine. Term recognition and extraction based on semantics for ontology construction. *International Journal of Computer Science*, 2012.

[11] Nguyen Bach and Sameer Badaskar. A review of relation extraction. In *Literature review for Language and Statistics II*, 2007.

[12] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *COLING-ACL*, pages 86–90. Morgan Kaufmann Publishers / ACL, 1998.

[13] Jason Baldridge. The OpenNLP project, 2010. URL: http://opennlp.apache.org/index.html, (accessed 2016-11-05).

[14] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[15] Roberto Bartolini, Emiliano Giovannetti, Simone Marchi, Simonetta Montemagni, Claudio Andreatta, Roberto Brunelli, Rodolfo Stecher, and Paolo Bouquet. Multimedia information extraction in ontology-based semantic annotation of product catalogues. In *Semantic Web Applications and Perspectives (SWAP)*. CEUR-WS.org, 2006.

[16] Sean Bechhofer, Yeliz Yesilada, Robert Stevens, Simon Jupp, and Bernard Horan. Using ontologies and vocabularies for dynamic linking. *IEEE Internet Computing*, 12(3):32–39, 2008.

[17] Adrian Benton and Mark Dredze. Entity linking for spoken language. In *Annual Conference of the North American Chapter of the ACL*, pages 225–230. The Association for Computational Linguistics, 2015.

[18] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1st edition, 1999.

[19] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel: Entity linking in web tables. In *International Semantic Web Conference (ISWC)*, pages 425–441. Springer, 2015.

[20] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. O'Reilly, 2009.

[21] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012.

[22] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[23] Eva Blomqvist. Ontocase – automatic ontology enrichment based on ontology design patterns. In *International Semantic Web Conference (ISWC)*, pages 65–80. Springer, 2009.

[24] Christoph Böhm, Gjergji Kasneci, and Felix Naumann. Latent topics in graph-structured data. In *Information and Knowledge Management (CIKM)*, pages 2663–2666. ACM, 2012.

[25] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *International Conference on Management of Data, SIGMOD*, pages 1247–1250, 2008.

[26] Johan Bos. Wide-coverage semantic analysis with boxer. In *Conference on Semantics in Text Processing, STEP*, pages 277–286. Association for Computational Linguistics, 2008.

[27] Eric Brill. A simple rule-based part of speech tagger. In *ANLP*, pages 152–155, 1992.

[28] Patrice Buche, Juliette Dibie-Barthélemy, Liliana Ibanescu, and Lydie Soler. Fuzzy web data tables integration guided by an ontological and terminological resource. *IEEE Trans. Knowl. Data Eng.*, 25(4):805–819, 2013.

[29] Paul Buitelaar and Bernardo Magnini. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12. IOS Press, 2005.

[30] Paul Buitelaar, Daniel Olejnik, and Michael Sintek. A protégé plug-in for ontology extraction from text based on linguistic analysis. In *European Semantic Web Symposium, ESWS*, pages 31–44. Springer, 2004.

[31] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *European Chapter of the Association for Computational Linguistics (EACL)*, pages 9–16, 2006.

[32] Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.

[33] Elena Cardillo, Josepph Roumier, Marc Jamoulle, and Robert Vander Stichele. Using ISO and Semantic Web standards for creating a Multilingual Medical Interface Terminology: A use case for Hearth Failure. *Terminology and Artificial Intelligence (TIA)*, 2013.

[34] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. Erd'14: entity recognition and disambiguation challenge. In *Conference on Research and Development in Information Retrieval, SIGIR*, page 1292. ACM, 2014.

[35] Bob Carpenter and Breck Baldwin. *Text Analysis with LingPipe 4*. LingPipe Publishing, 2011.

[36] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter: an open source framework for entity linking. In *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, pages 17–20. ACM, 2013.

[37] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter 2.0 - an Open Source Tool for Semantically Enriching Data. In *ISWC-PD*, pages 417–420. CEUR-WS.org, 2014.

[38] Mohamed Chabchoub, Michel Gagnon, and Amal Zouaq. Collective disambiguation and semantic annotation for entity linking and typing. In *Extended Semantic Web Conference (ESWC)*, pages 33–47. Springer, 2016.

[39] Eric Charton, Michel Gagnon, and Benoît Ozell. Automatic semantic web annotation of named entities. In *Canadian Conference on Artificial Intelligence*, pages 74–85. Springer Berlin Heidelberg, 2011.

[40] Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. Modeling Documents by Combining Semantic Concepts with Unsupervised Statistical Learning. In *International Semantic Web Conference (ISWC)*, pages 229–244. Springer, 2008.

[41] Danqi Chen and Christopher D. Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014.

[42] Hsin-Hsi Chen, Shih-Chung Tsai, and Jin-He Tsai. Mining tables from large scale HTML texts. In *International Conference on Computational Linguistics (COLING)*, pages 166–172. Morgan Kaufmann, 2000.

[43] Long Chen, Joemon M Jose, Haitao Yu, Fajie Yuan, and Huaizhi Zhang. Probabilistic topic modelling with semantic graph. In *European Conference on Information Retrieval (ECIR)*, pages 240–251. Springer, 2016.

[44] Yen-Pin Chiu, Yong-Siang Shih, Yang-Yin Lee, Chih-Chieh Shao, Ming-Lun Cai, Sheng-Lun Wei, and Hsin-Hsi Chen. NTUNLP approaches to recognizing and disambiguating entities in long and short text at the ERD challenge 2014. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 3–12. ACM, 2014.

[45] Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. Two decades of unsupervised POS induction: How far have we come? In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 575–584, 2010.

[46] Philipp Cimiano. *Ontology learning and population from text - algorithms, evaluation and applications*. Springer, 2006.

[47] Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *World Wide Web Conference (WWW)*, pages 462–471. ACM, 2004.

[48] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Comparing conceptual, divise and agglomerative clustering for learning taxonomies from text. In *European Conference on Artificial Intelligence (ECAI)*, pages 435–439. IOS Press, 2004.

[49] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *J. Artif. Intell. Res.*, 24:305–339, 2005.

[50] Philipp Cimiano, John P. McCrae, and Paul Buitelaar. Lexicon model for ontologies: Community report. W3C Final Community Group Report, May 2016.

[51] Philipp Cimiano, John P McCrae, Víctor Rodríguez-Doncel, Tatiana Gornostay, Asunción Gómez-Pérez, Benjamin Siemoneit, and Andis Lagzdins. Linked terminology: Applying linked data principles to terminological resources. *eLex*, 2015.

[52] Philipp Cimiano and Johanna Völker. Text2onto: A framework for ontology learning and data-driven change discovery. In *International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 227–238. Springer, 2005.

[53] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2256–2262, 2016.

[54] Francesco Colace, Massimo De Santo, Luca Greco, Vincenzo Moscato, and Antonio Picariello. Probabilistic approaches for sentiment analysis: Latent dirichlet allocation for ontology building and sentiment extraction. In *Sentiment Analysis and Ontology Engineering - An Environment of Computational Intelligence*, pages 75–91. Springer, 2016.

[55] Michael Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with Perceptron al-

gorithms. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8. Association for Computational Linguistics, 2002.

[56] Angel Conde, Mikel Larrañaga, Ana Arruarte, Jon A Elorriaga, and Dan Roth. litewi: A combined term extraction and entity linking method for eliciting educational ontologies from textbooks. *Journal of the Association for Information Science and Technology*, 67(2):380–399, 2016.

[57] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *World Wide Web Conference (WWW)*, 2013.

[58] Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze. A Piggyback System for Joint Entity Mention Detection and Linking in Web Queries. In *World Wide Web Conference (WWW)*, pages 567–578. ACM, 2016.

[59] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *World Wide Web Conference (WWW)*, pages 355–366. ACM, 2013.

[60] Kino Coursey, Rada Mihalcea, and William E. Moen. Using encyclopedic knowledge for automatic topic identification. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 210–218. Association for Computational Linguistics, 2009.

[61] Eric Crestan and Patrick Pantel. Web-scale table census and classification. In *Web Search and Web Data Mining (WSDM)*, pages 545–554. ACM, 2011.

[62] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–716. Association for Computational Linguistics, 2007.

[63] Silviu Cucerzan. Name entities made obvious: the participation in the ERD 2014 evaluation. In *Workshop on Entity Recognition & Disambiguation, ERD*, pages 95–100. ACM, 2014.

[64] Hamish Cunningham. GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[65] James R. Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale NLP with c&c and boxer. In *Annual meeting of the Association for Computational Linguistics (ACL)*. The Association for Computational Linguistics, 2007.

[66] Merley da Silva Conrado, Ariani Di Felippo, Thiago Alexandre Salgueiro Pardo, and Solange Oliveira Rezende. A survey of automatic term extraction for brazilian portuguese. *Journal of the Brazilian Computer Society*, 20(1):1–28, 2014.

[67] Jan Daciuk, Stoyan Mihov, Bruce W. Watson, and Richard Watson. Incremental Construction of Minimal Acyclic Finite State Automata. *Computational Linguistics*, 26(1):3–16, 2000.

[68] Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF Mapping Language. W3C Recommendation, September 2012.

[69] Davor Delac, Zoran Krleza, Jan Snajder, Bojana Dalbelo Basic, and Frane Saric. TermeX: A Tool for Collocation Extraction. In *Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 149–157. Springer, 2009.

[70] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *World Wide Web Conference (WWW)*, pages 469–478. ACM, 2012.

[71] Leon Derczynski, Isabelle Augenstein, and Kalina Bontcheva. USFD: twitter NER with drift compensation and linked data. *CoRR*, abs/1511.03088, 2015.

[72] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32 – 49, 2015.

[73] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[74] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, Ramanathan V. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *World Wide Web Conference (WWW)*, pages 178–186. ACM, 2003.

[75] Li Ding, Dominic DiFranzo, Alvaro Graves, James Michaelis, Xian Li, Deborah L. McGuinness, and Jim Hendler. Data-gov wiki: Towards linking government data. In *Linked Data Meets Artificial Intelligence, AAAI*. AAAI, 2010.

[76] Milan Dojchinovski and Tomáš Kliegr. Recognizing, classifying and linking entities with Wikipedia and DBpedia. In *Workshop on Intelligent and Knowledge Oriented Technologies (WIKT)*, pages 41–44, 2012.

[77] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Edith Schonberg, and Kavitha Srinivas. Extracting enterprise vocabularies using linked open data. In *International Semantic Web Conference (ISWC)*, pages 779–794. Springer, 2009.

[78] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 601–610. ACM, 2014.

[79] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *International Conference on Computational Linguistics (COLING)*, pages 277–285. Tsinghua University Press, 2010.

[80] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[81] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490, 2014.

[82] Arnab Dutta, Christian Meilicke, and Heiner Stuckenschmidt. Semantifying triples from open information extraction systems. In *STAIRS*, pages 111–120. IOS Press, 2014.

[83] Arnab Dutta, Christian Meilicke, and Heiner Stuckenschmidt. Enriching structured knowledge with open information. In *World Wide Web Conference (WWW)*, pages 267–277. ACM, 2015.

[84] Martin Dzbor, Enrico Motta, and John Domingue. Magpie: Experiences in supporting semantic web browsing. *J. Web Sem.*, 5(3):204–222, 2007.

[85] Jay Earley. An Efficient Context-Free Parsing Algorithm. *Commun. ACM*, 13(2):94–102, 1970.

[86] Alan Eckhardt, Juraj Hresko, Jan Procházka, and Otakar Smrs. Entity linking based on the co-occurrence graph

and entity probability. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 37–44. ACM, 2014.

[87] Peter Exner and Pierre Nugues. Entity extraction: From unstructured text to DBpedia RDF triples. In *The Web of Linked Entities Workshop (WoLE 2012)*, pages 58–69. CEUR-WS, 2012.

[88] Peter Exner and Pierre Nugues. Refractive: an open source tool to extract knowledge from Syntactic and Semantic Relations. In *Language Resources and Evaluation Conference (LREC)*. European Language Resources Association (ELRA), 2014.

[89] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545. ACL, 2011.

[90] Ángel Felices-Lago and Pedro Ureña Gómez-Moreno. FunGramKB term extractor: A tool for building terminological ontologies from specialised corpora. In *Studies in Language Companion Series*, pages 251–270. John Benjamins Publishing Company, 2014.

[91] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Information and Knowledge Management (CIKM)*, pages 1625–1628, New York, NY, USA, 2010. ACM.

[92] David A. Ferrucci and Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[93] Charles J Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.

[94] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 363–370. Association for Computational Linguistics, 2005.

[95] Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 141–150. ACL, 2009.

[96] Marco Fossati, Emilio Dorigatti, and Claudio Giuliano. N-ary relation extraction for simultaneous t-box and a-box knowledge base augmentation. *Semantic Web Journal*, 2017.

[97] Katerina T. Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms: the c-value/nc-value method. *Int. J. on Digital Libraries*, 3(2):115–130, 2000.

[98] André Freitas, Danilo S Carvalho, João CP Da Silva, Seán O'Riain, and Edward Curry. A semantic best-effort approach for extracting structured discourse graphs from Wikipedia. In *Workshop on the Web of Linked Entities, (ISWC-WLE)*, 2012.

[99] David S Friedlander. *Semantic Information Extraction*. CRC Press, Boca Raton, 2005.

[100] Michel Gagnon, Amal Zouaq, and Ludovic Jean-Louis. Can we use linked data semantic annotators for the extraction of domain-relevant expressions? In *World Wide Web Conference (WWW)*, pages 1239–1246. ACM, 2013.

[101] Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In *Extended Semantic Web Conference (ESWC)*, pages 351–366. Springer, 2013.

[102] Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovì. Semantic Web Machine Reading with FRED. *Semantic Web*, 8(6):873–893, 2017.

[103] Aldo Gangemi, Diego Reforgiato Recupero, Misael Mongiovì, Andrea Giovanni Nuzzolese, and Valentina Presutti. Identifying motifs for evaluating open knowledge extraction on the web. *Knowl.-Based Syst.*, 108:33–41, 2016.

[104] Ning Gao and Silviu Cucerzan. Entity linking to one thousand knowledge bases. In *European Conference on IR Research (ECIR)*, pages 1–14. Springer, 2017.

[105] Anna Lisa Gentile, Ziqi Zhang, and Fabio Ciravegna. Self training wrapper induction with linked data. In *Text, Speech and Dialogue (TSD)*, pages 285–292. Springer, 2014.

[106] Daniel Gerber, Sebastian Hellmann, Lorenz Bühmann, Tommaso Soru, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Real-Time RDF Extraction from Unstructured Data Streams. In *International Semantic Web Conference (ISWC)*, volume 8218, pages 135–150. Springer Berlin Heidelberg, 2013.

[107] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Extracting Multilingual Natural-language Patterns for RDF Predicates. In *Knowledge Engineering and Knowledge Management (EKAW)*, pages 87–96. Springer-Verlag, 2012.

[108] Silvia Giannini, Simona Colucci, Francesco M. Donini, and Eugenio Di Sciascio. A Logic-Based Approach to Named-Entity Disambiguation in the Web of Data. In *Advances in Artificial Intelligence*, pages 367–380. Springer, 2015.

[109] Lee Gillam, Mariam Tariq, and Khurshid Ahmad. Terminology and the construction of ontology. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 11(1):55–81, 2005.

[110] Giuseppe Rizzo and Raphaël Troncy. NERD: evaluating named entity recognition tools in the web of data. In *ISWC-WEKEX*, 2011.

[111] Michel L. Goldstein, Steve A. Morris, and Gary G. Yen. Bridging the gap between data acquisition and inference ontologies–towards ontology based link discovery. In *SPIE*, volume 5071, page 117, 2003.

[112] Toni Grütze, Gjergji Kasneci, Zhe Zuo, and Felix Naumann. CohEEL: Coherent and efficient named entity linking through random walks. *J. Web Sem.*, 37-38:75–89, 2016.

[113] Jon Atle Gulla, Hans Olaf Borch, and Jon Espen Ingvaldsen. Unsupervised keyphrase extraction for search ontologies. In *International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 25–36. Springer, 2006.

[114] Sherzod Hakimov, Salih Atilay Oto, and Erdogan Dogdu. Named entity recognition and disambiguation using linked data and graph-based centrality scoring. In *International Workshop on Semantic Web Information Management (SWIM)*, pages 4:1–4:7. ACM, 2012.

[115] Sherzod Hakimov, Hendrik ter Horst, Soufian Jebbara, Matthias Hartung, and Philipp Cimiano. Combining Textual and Graph-Based Features for Named Entity Disambiguation Using Undirected Probabilistic Graphical Models. In *Knowledge Engineering and Knowledge Management (EKAW)*, pages 288–302. Springer, 2016.

[116] Mostafa M. Hassan, Fakhri Karray, and Mohamed S. Kamel. Automatic document topic identification using Wikipedia hierarchical ontology. In *Information Science, Signal Process-*

ing and their Applications (ISSPA), pages 237–242. IEEE, 2012.

[117] Austin Haugen. Abstract: The open graph protocol design decisions. In *International Semantic Web Conference (ISWC)*, page 338. Springer, 2010.

[118] David G. Hays. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525, 1964.

[119] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *International Conference on Computational Linguistics (COLING)*, pages 539–545, 1992.

[120] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using linked data. In *International Semantic Web Conference (ISWC)*, pages 98–113. Springer, 2013.

[121] Martin Hepp. GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In *Knowledge Engineering and Knowledge Management (EKAW)*, pages 329–346. Springer, 2008.

[122] Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *Annual meeting of the Association for Computational Linguistics (ACL)*, 2000.

[123] Daniel Hernández, Aidan Hogan, and Markus Krötzsch. Reifying rdf: What works well with wikidata? In *International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS)*, page 32, 2015.

[124] Timm Heuss, Bernhard Humm, Christian Henninger, and Thomas Rippl. A comparison of NER tools w.r.t. a domain-specific vocabulary. In *International Conference on Semantic Systems (SEMANTICS)*, pages 100–107. ACM, 2014.

[125] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. Owl 2 web ontology language primer (second edition). W3C Recommendation, December 2012.

[126] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In *World Wide Web Conference (WWW)*, pages 385–396. ACM, 2014.

[127] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. KORE: keyphrase overlap relatedness for entity disambiguation. In *Information and Knowledge Management (CIKM)*, pages 545–554. ACM, 2012.

[128] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[129] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792. Association for Computational Linguistics, 2011.

[130] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 541–550. ACL, 2011.

[131] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[132] Yinghao Huang, Xipeng Wang, and Yi Lu Murphey. Text categorization using topic model and ontology networks. In *International Conference on Data Mining (DMIN)*, 2014.

[133] Ioana Hulpuş, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Web Search and Web Data Mining (WSDM)*, pages 465–474. ACM, 2013.

[134] Ioana Hulpuş, Narumol Prangnawarat, and Conor Hayes. Path-Based Semantic Relatedness on Linked Data and Its Use to Word and Entity Disambiguation. In *International Semantic Web Conference (ISWC)*, pages 442–457. Springer, 2015.

[135] Dat T Huynh, Tru H Cao, Phuong HT Pham, and Toan N Hoang. Using hyperlink texts to improve quality of identifying document topics based on wikipedia. In *International Conference on Knowledge and Systems Engineering (KSE)*, pages 249–254. IEEE, 2009.

[136] David Huynh, Stefano Mazzocchi, and David R. Karger. Piggy Bank: Experience the Semantic Web inside your web browser. *J. Web Sem.*, 5(1):16–27, 2007.

[137] Uraiwan Inyaem, Phayung Meesad, Choochart Haruechaiyasak, and Dat Tran. Construction of fuzzy ontology-based terrorism event extraction. In *International Conference on Knowledge Discovery and Data Mining (WKDD)*, pages 391–394. IEEE, 2010.

[138] Sonal Jain and Jyoti Pareek. Automatic topic(s) identification from learning material: An ontological approach. In *Computer Engineering and Applications (ICCEA)*, volume 2, pages 358–362. IEEE, 2010.

[139] Maciej Janik and Krys Kochut. Wikipedia in action: Ontological knowledge in text categorization. In *International Conference on Semantic Computing (ICSC)*, pages 268–275. IEEE Computer Society, 2008.

[140] Ludovic Jean-Louis, Amal Zouaq, Michel Gagnon, and Faezeh Ensan. An assessment of online semantic annotators for the keyword extraction task. In *Pacific Rim International Conference on Artificial Intelligence PRICAI*, pages 548–560. Springer, 2014.

[141] Xing Jiang and Ah-Hwee Tan. CRCTOL: A semantic-based domain ontology learning system. *JASIST*, 61(1):150–168, 2010.

[142] Jelena Jovanovic, Ebrahim Bagheri, John Cuzzola, Dragan Gasevic, Zoran Jeremic, and Reza Bashash. Automated Semantic Tagging of Textual Content. *IT Professional*, 16(6):38–46, nov 2014.

[143] Yutaka Kabutoya, Róbert Sumi, Tomoharu Iwata, Toshio Uchiyama, and Tadasu Uchiyama. A topic model for recommending movies via linked open data. In *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 625–630. IEEE, 2012.

[144] Hans Kamp. A theory of truth and semantic representation. In P. Portner and B. H. Partee, editors, *Formal Semantics – the Essential Readings*, pages 189–222. Blackwell, 1981.

[145] Fred Karlsson, Atro Voutilainen, Juha Heikkilae, and Arto Anttila. *Constraint Grammar: a language-independent system for parsing unrestricted text*, volume 4. Walter de Gruyter, 1995.

[146] Steffen Kemmerer, Benjamin Großmann, Christina Müller, Peter Adolphs, and Heiko Ehrig. The Neofonie NERD system at the ERD challenge 2014. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 83–88. ACM, 2014.

[147] Ali Khalili, Sören Auer, and Daniel Hladky. The rdfa content editor - from WYSIWYG to WYSIWYM. In *Computer Software and Applications Conference (COMPSAC)*, pages 531–540. IEEE Computer Society, 2012.

[148] Hak Lae Kim, Simon Scerri, John G. Breslin, Stefan Decker, and Hong-Gee Kim. The state of the art in tag ontologies: A semantic model for tagging and folksonomies. In *International Conference on Dublin Core and Metadata Applications (DC)*, pages 128–137, 2008.

[149] Jung-jae Kim and Dietrich Rebholz-Schuhmann. Improving the extraction of complex regulatory events from scientific text by using ontology-based inference. *J. Biomedical Semantics*, 2(S-5):S3, 2011.

[150] Jung-jae Kim and Luu Anh Tuan. Hybrid pattern matching for complex ontology term recognition. In *Conference on Bioinformatics, Computational Biology and Biomedicine (BCB)*, pages 289–296. ACM, 2012.

[151] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *International Workshop on Semantic Evaluation (SemEval)*, pages 21–26. Association for Computational Linguistics, 2010.

[152] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Language Resources and Evaluation Conference (LREC)*. European Language Resources Association, 2002.

[153] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extending VerbNet with Novel Verb Classes. In *Language Resources and Evaluation Conference (LREC)*, pages 1027–1032. European Language Resources Association (ELRA), 2006.

[154] Bettina Klimek, John P. McCrae, Christian Lehmann, Christian Chiarcos, and Sebastian Hellmann. OnLiT: An Ontology for Linguistic Terminology. In *International Conference on Language, Data, and Knowledge (LDK)*, pages 42–57. Springer, 2017.

[155] Sebastian Krause, Leonhard Hennig, Andrea Moro, Dirk Weissenborn, Feiyu Xu, Hans Uszkoreit, and Roberto Navigli. Sar-graphs: A language resource connecting linguistic knowledge with semantic relations from knowledge graphs. *J. Web Sem.*, 37-38:112–131, 2016.

[156] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[157] Javier Lacasta, Javier Nogueras Iso, and Francisco Javier Zarazaga Soria. *Terminological Ontologies*. Springer US, 2010.

[158] Anne Lauscher, Federico Nanni, Pablo Ruiz Fabo, and Simone Paolo Ponzetto. Entities as topic labels: combining entity linking and labeled lda to improve topic interpretability and evaluability. *Italian Journal of Computational Linguistics*, 2(2):67–88, 2016.

[159] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[160] Oliver Lehmberg, Dominique Ritze, Petar Ristoski, Robert Meusel, Heiko Paulheim, and Christian Bizer. The Mannheim Search Join Engine. *J. Web Sem.*, 35:159–166, 2015.

[161] Lothar Lemnitzer, Cristina Vertan, Alex Killing, Kiril Ivanov Simov, Diane Evans, Dan Cristea, and Paola Monachesi. Improving the search for learning objects with keywords and ontologies. In *European Conference on Technology Enhanced Learning (EC-TEL)*, pages 202–216. Springer, 2007.

[162] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[163] Yuncheng Li, Xitong Yang, and Jiebo Luo. Semantic video entity linking based on visual content and metadata. In *International Conference on Computer Vision (ICCV)*, pages 4615–4623. IEEE Computer Society, 2015.

[164] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and Searching Web Tables Using Entities, Types and Relationships. *PVLDB*, 3(1):1338–1347, 2010.

[165] Chin-Yew Lin. Knowledge-based automatic topic identification. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 308–310. Association for Computational Linguistics, 1995.

[166] Dekang Lin and Patrick Pantel. Dirt — sbt discovery of inference rules from text. In *International conference on Knowledge discovery and data mining (SIGKDD)*, pages 323–328. ACM, 2001.

[167] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural Relation Extraction with Selective Attention over Instances. In *Association for Computational Linguistics (ACL)*. ACL, 2016.

[168] Marek Lipczak, Arash Koushkestani, and Evangelos E. Milios. Tulip: lightweight entity recognition and disambiguation using Wikipedia-based topic centroids. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 31–36, 2014.

[169] Fang Liu, Shizhu He, Shulin Liu, Guangyou Zhou, Kang Liu, and Jun Zhao. Open relation mapping based on instances and semantics expansion. In *Asia Information Retrieval Societies Conference (AIRS)*, pages 320–331. Springer, 2013.

[170] Wei Lu and Dan Roth. Joint mention extraction and classification with mention hypergraphs. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 857–867. The Association for Computational Linguistics, 2015.

[171] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint Entity Recognition and Disambiguation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 879–888. The Association for Computational Linguistics, 2015.

[172] Lieve Macken, Els Lefever, and Veronique Hoste. TExSIS: Bilingual Terminology Extraction from Parallel Corpora Using Chunk-based Alignment. *Terminology*, 19(1):1–30, 2013.

[173] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, March 2001.

[174] Tiep Mai, Bichen Shi, Patrick K. Nicholson, Deepak Ajwani, and Alessandra Sala. Distributed entity disambiguation with per-mention learning. *CoRR*, abs/1604.05875, 2016.

[175] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 55–60, 2014.

[176] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[177] Luís Marujo, Anatole Gershman, Jaime G. Carbonell, Robert E. Frederking, and João Paulo Neto. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In *Language Resources and Evaluation Conference (LREC)*, 2012.

[178] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Empirical Methods in Natural Language Processing (EMNLP) and (CoNLL)*, pages 523–534. ACL, 2012.

[179] Diana Maynard, Kalina Bontcheva, and Isabelle Augenstein. *Natural Language Processing for the Semantic Web*. Morgan & Claypool, 2016.

[180] Jon D. Mcauliffe and David M. Blei. Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128. Curran Associates, Inc., 2008.

[181] Joseph F. McCarthy and Wendy G. Lehnert. Using Decision Trees for Coreference Resolution. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1050–1055, 1995.

[182] John P. McCrae, Steven Moran, Sebastian Hellmann, and Martin Brümmer. Multilingual linked data. *Semantic Web*, 6(4):315–317, 2015.

[183] Alyona Medelyan. Nlp keyword extraction tutorial with rake and maui. online, 2014.

[184] Olena Medelyan, Steve Manion, Jeen Broekstra, Anna Divoli, Anna Lan Huang, and Ian Witten. Constructing a focused taxonomy from a document collection. In *Extended Semantic Web Conference (ESWC)*, 2013.

[185] Olena Medelyan, Ian H Witten, and David Milne. Topic indexing with wikipedia. In *Wikipedia and Artificial Intelligence: An Evolving Synergy*, page 19, 2008.

[186] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. In *Web Search and Web Data Mining (WSDM)*, pages 563–572. ACM, 2012.

[187] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *International Conference on Semantic Systems (I-Semantics)*, pages 1–8, New York, NY, USA, 2011. ACM.

[188] Robert Meusel, Christian Bizer, and Heiko Paulheim. A web-scale study of the adoption and evolution of the schema.org vocabulary over time. In *Web Intelligence, Mining and Semantics (WIMS)*, pages 15:1–15:11, 2015.

[189] Robert Meusel, Petar Petrovski, and Christian Bizer. The WebDataCommons microdata, RDFa and microformat dataset series. In *International Semantic Web Conference (ISWC)*, pages 277–292. Springer, 2014.

[190] Rada Mihalcea and Andras Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Information and Knowledge Management (CIKM)*, pages 233–242. ACM, 2007.

[191] Pavel Mihaylov and Davide Palmisano. D4.5 integration of advanced modules in the annotation framework. Deliverable of the NoTube FP7 EU project (project no. 231761), 2011. Available at `http://notube3.files.wordpress.com/2012/01/notube_d4-5-integration-of-advanced-modules-in-annotation-framework-vm33.pdf`.

[192] Peter Mika. On Schema.org and Why It Matters for the Web. *IEEE Internet Computing*, 19(4):52–55, 2015.

[193] Peter Mika, Edgar Meij, and Hugo Zaragoza. Investigating the semantic gap through query log analysis. In *International Semantic Web Conference (ISWC)*, pages 441–455. Springer, 2009.

[194] Alistair Miles and Sean Bechhofer. Skos simple knowledge organization system reference. W3C Recommendation, August 2009.

[195] George A. Miller. WordNet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[196] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Information and Knowledge Management (CIKM)*, pages 509–518. ACM, 2008.

[197] David N. Milne and Ian H. Witten. An open-source toolkit for mining Wikipedia. *Artif. Intell.*, 194:222–239, 2013.

[198] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *North American Chapter of the (ACL)*, pages 777–782. The Association for Computational Linguistics, 2013.

[199] Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. Meantime, the newsreader multilingual event and time corpus. In *Language Resources and Evaluation Conference (LREC)*. European Language Resources Association (ELRA), 2016.

[200] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 1003–1011, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[201] Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-Ending Learning. In *Conference on Artificial Intelligence (AAAI)*, pages 2302–2310. AAAI Press, 2015.

[202] Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, and Boi Faltings. Keyword extraction from the web for foaf metadata. In *Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[203] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[204] Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In *International Semantic Web Conference (ISWC)*, pages 363–378. Springer, 2013.

[205] Emir Muñoz, Aidan Hogan, and Alessandra Mileo. Using linked data to mine RDF from wikipedia's tables. In *Web Search and Web Data Mining (WSDM)*, pages 533–542. ACM, 2014.

[206] Oscar Muñoz-García, Andres García-Silva, Oscar Corcho, Manuel de la Higuera-Hernández, and Carlos Navarro. Identifying topics in social media posts using DBpedia. In *Net-*

worked and Electronic Media Summit (NEM), 2011.

[207] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[208] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable knowledge harvesting with high precision and high recall. In *Web Search and Web Data Mining (WSDM)*, pages 227–236. ACM, 2011.

[209] Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. Discovering semantic relations from the web and organizing them with PATTY. *SIGMOD Record*, 42(2):29–34, 2013.

[210] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A semantic metadata generator for web pages based on keyphrase extraction. In *ISWC-PD*, pages 201–204. CEUR-WS.org, 2014.

[211] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, 2009.

[212] Roberto Navigli, Paola Velardi, and Aldo Gangemi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1):22–31, 2003.

[213] Kamel Nebhi. A Rule-based Relation Extraction System Using DBpedia and Syntactic Parsing. In *Conference on NLP & DBpedia (NLP-DBPEDIA)*, pages 74–79, Aachen, Germany, Germany, 2013. CEUR-WS.org.

[214] Claire Nedellec, Wiktoria Golik, Sophie Aubin, and Robert Bossy. Building large lexicalized ontologies from text: a use case in automatic indexing of biotechnology patents. In *Knowledge Engineering and Knowledge Management (EKAW)*, pages 514–523. Springer, 2010.

[215] Gerald Nelson, Sean Wallis, and Bas Aarts. *Exploring natural language: working with the British component of the International Corpus of English*, volume 29. John Benjamins Publishing, 2002.

[216] Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, René Speck, and Martin Kaltenböck. SCMS - semantifying content management systems. In *International Semantic Web Conference (ISWC)*, pages 189–204. Springer, 2011.

[217] Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. Aida-light: High-throughput named-entity disambiguation. In *World Wide Web Conference (WWW)*. CEUR-WS.org, 2014.

[218] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-NERD: joint named entity recognition and disambiguation with rich linguistic features. *TACL*, 4:215–229, 2016.

[219] Truc-Vien T. Nguyen and Alessandro Moschitti. End-to-end relation extraction using distant supervision from external semantic repositories. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 277–282. Association for Computational Linguistics, 2011.

[220] Feng Niu, Ce Zhang, Christopher Ré, and Jude W. Shavlik. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. In *International Workshop on Searching and Integrating New Web Data Sources*, pages 25–28. CEUR-WS.org, 2012.

[221] Joakim Nivre. Dependency parsing. *Language and Linguistics Compass*, 4(3):138–152, 2010.

[222] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Language Resources and Evaluation Conference (LREC)*, 2016.

[223] Vít Novácek, Loredana Laera, Siegfried Handschuh, and Brian Davis. Infrastructure for dynamic knowledge integration - automated biomedical ontology extension using textual resources. *Journal of Biomedical Informatics*, 41(5):816–828, 2008.

[224] Bernardo Pereira Nunes, Stefan Dietze, Marco Antonio Casanova, Ricardo Kawase, Besnik Fetahu, and Wolfgang Nejdl. Combining a co-occurrence-based and a semantic measure for entity linking. In *Extended Semantic Web Conference (ESWC)*, pages 548–562. Springer, 2013.

[225] Alex Olieman, Hosein Azarbonyad, Mostafa Dehghani, Jaap Kamps, and Maarten Marx. Entity linking by focusing DBpedia candidate entities. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 13–24. ACM, 2014.

[226] Hanane Ouksili, Zoubida Kedad, and Stéphane Lopes. Theme identification in rdf graphs. In *International Conference on Model and Data Engineering (MEDI)*, pages 321–329. Springer, 2014.

[227] Rifat Ozcan and YA Aslangdogan. Concept based information access using ontologies and latent semantic analysis. *Dept. of Computer Science and Engineering*, 8:2004, 2004.

[228] Maria Pazienza, Marco Pennacchiotti, and Fabio Zanzotto. Terminology extraction: an analysis of linguistic and statistical approaches. *Knowledge mining*, pages 255–279, 2005.

[229] Francesco Piccinno and Paolo Ferragina. From TagME to WAT: A New Entity Annotator. In *International Workshop on Entity Recognition & Disambiguation (ERD)*, pages 55–62. ACM, 2014.

[230] Pablo Pirnay-Dummer and Satjawan Walter. Bridging the world's knowledge to individual knowledge using latent semantic analysis and web ontologies to complement classical and new knowledge assessment technologies. *Technology, Instruction, Cognition & Learning*, 7(1), 2009.

[231] Aleksander Pivk, Philipp Cimiano, York Sure, Matjaz Gams, Vladislav Rajkovic, and Rudi Studer. Transforming arbitrary tables into logical form with TARTAR. *Data Knowl. Eng.*, 60(3):567–595, 2007.

[232] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. A hybrid approach for entity recognition and linking. In *ESWC-SemWebEval*, pages 28–39. Springer, 2015.

[233] Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. Enhancing entity linking by combining ner models. In *Extended Semantic Web Conference (ESWC)*, 2016.

[234] Axel Polleres, Aidan Hogan, Andreas Harth, and Stefan Decker. Can we ever catch up with the web? *Semantic Web*, 1(1-2):45–52, 2010.

[235] Hoifung Poon and Pedro M. Domingos. Joint Unsupervised Coreference Resolution with Markov Logic. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 650–659, 2008.

[236] Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. KIM - a semantic platform for information extraction and retrieval. *Natural Language Engineering*, 10(3-4):375–392, 2004.

[237] Valentina Presutti, Andrea Giovanni Nuzzolese, Sergio Consoli, Aldo Gangemi, and Diego Reforgiato Recupero. From hyperlinks to semantic web properties using open knowledge extraction. *Semantic Web*, 7(4):351–378, 2016.

[238] Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, Felice Ferrara, and Carlo Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *Int. J. Intell. Syst.*, 25(12):1158–1186, December 2010. TE.

[239] Yves Raimond, Tristan Ferne, Michael Smethurst, and Gareth Adams. The BBC World Service Archive prototype. *J. Web Sem.*, 27:2–9, 2014.

[240] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 1375–1384. The Association for Computer Linguistics, 2011.

[241] Adwait Ratnaparkhi. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175, 1999.

[242] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases PKDD*, pages 148–163. Springer, 2010.

[243] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation Extraction with Matrix Factorization and Universal Schemas. In *Association of Computational Linguistics (ACL)*, pages 74–84. ACL, 2013.

[244] Sebastián A. Ríos, Felipe Aguilera, Francisco Bustos, Tope Omitola, and Nigel Shadbolt. Leveraging social network analysis with topic models and the semantic web. In *Web Intelligence and Intelligent Agent Technology*, pages 339–342. IEEE Computer Society, 2011.

[245] Ana B. Rios-Alvarado, Ivan López-Arévalo, and Víctor Jesús Sosa Sosa. Learning concept hierarchies from textual resources for ontologies construction. *Expert Systems with Applications*, 40(15):5907–5915, 2013.

[246] Petar Ristoski and Heiko Paulheim. Semantic Web in data mining and knowledge discovery: A comprehensive survey. *J. Web Sem.*, 36:1–22, 2016.

[247] Dominique Ritze and Christian Bizer. Matching Web Tables To DBpedia – A Feature Utility Study. In *International Conference on Extending Database Technology (EDBT)*, pages 210–221. OpenProceedings.org, 2017.

[248] Giuseppe Rizzo and Raphaël Troncy. NERD: A framework for unifying named entity recognition and disambiguation extraction tools. In *European Chapter of the (ACL)*, pages 73–76. The Association for Computer Linguistics, 2012.

[249] Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In *Language Resources and Evaluation Conference (LREC)*, Reykjavik, ISLANDE, 05 2014.

[250] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Web Search and Web Data Mining (WSDM)*, pages 399–408. ACM, 2015.

[251] Michael Röder, Axel-Cyrille Ngonga Ngomo, Ivan Ermilov, and Andreas Both. Detecting similar linked datasets using topic modelling. In *Extended Semantic Web Conference (ESWC)*, pages 3–19. Springer, 2016.

[252] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[253] Jacobo Rouces, Gerard de Melo, and Katja Hose. Framebase: Representing n-ary relations using semantic frames. In *Extended Semantic Web Conference (ESWC)*, pages 505–521. Springer, 2015.

[254] David Sánchez and Antonio Moreno. Learning medical ontologies from the web. *Knowledge Management for Health Care Procedures*, pages 32–45, 2008.

[255] Sunita Sarawagi. Information extraction. *Found. Trends databases*, 1(3):261–377, March 2008.

[256] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In *International Semantic Web Conference (ISWC)*, pages 245–260. Springer, 2014.

[257] Thomas Schmidt. The kicktionary revisited. In *(KONVENS)*, pages 239–251. Mouton de Gruyter, 2008.

[258] Peter Schönhofen. Identifying document topics using the Wikipedia category network. *Web Intelligence and Agent Systems: An International Journal*, 7(2):195–207, 2009.

[259] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. LIEGE: : link entities in web lists with knowledge base. In *Knowledge Discovery and Data Mining (KDD)*, pages 1424–1432. ACM, 2012.

[260] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In *World Wide Web Conference (WWW)*, pages 449–458. ACM, 2012.

[261] Amit Sheth, I Budak Arpinar, and Vipul Kashyap. Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships. In *Enhancing the Power of the Internet*, pages 63–94. Springer, 2004.

[262] Sifatullah Siddiqi and Aditi Sharan. Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications*, 109(2), 2015.

[263] Avirup Sil and Alexander Yates. Re-ranking for joint named-entity recognition and linking. In *Information and Knowledge Management (CIKM)*, pages 2369–2374. ACM, 2013.

[264] Abhishek SinghRathore and Devshri Roy. Ontology based web page topic identification. *International Journal of Computer Applications*, 85(6):35–40, 2014.

[265] Jennifer Sleeman, Tim Finin, and Anupam Joshi. Topic Modeling for RDF Graphs. In *ISWC-LD4IE*. CEUR-WS.org, 2015.

[266] Anton Södergren. HERD – Hajen Entity Recognition and Disambiguation, 2016.

[267] Stephen Soderland and Bhushan Mandhani. Moving from textual relations to ontologized relations. In *AAAI*, 2007.

[268] Wee Meng Soon, Hwee Tou Ng, and Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[269] Lucia Specia and Enrico Motta. Integrating folksonomies with the semantic web. In *European Semantic Web Conference (ESWC)*, pages 624–639. Springer, 2007.

[270] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *International Semantic Web Conference (ISWC)*, pages 519–534. Springer, 2014.

[271] Jian-Tao Sun, Zheng Chen, Hua-Jun Zeng, Yuchang Lu, Chun-Yi Shi, and Wei-Ying Ma. Supervised latent semantic indexing for document categorization. In *International Conference on Data Mining (ICDM)*, pages 535–538. IEEE, 2004.

[272] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In *Empirical Methods in Natural*

Language Processing (EMNLP), EMNLP-CoNLL '12, pages 455–465, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[273] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing Wrong Labels in Distant Supervision for Relation Extraction. In *Association for Computational Linguistics (ACL)*, pages 721–729. ACL, 2012.

[274] Thomas Pellissier Tanon, Denny Vrandecic, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From Freebase to Wikidata: The Great Migration. In *World Wide Web Conference (WWW)*, pages 1419–1428. ACM, 2016.

[275] Philippe Thomas, Johannes Starlinger, Alexander Vowinkel, Sebastian Arzt, and Ulf Leser. Geneview: a comprehensive semantic search engine for pubmed. *Nucleic acids research*, 40(W1):W585–W591, 2012.

[276] Sabrina Tiun, Rosni Abdullah, and Tang Enya Kong. Automatic topic identification using ontology hierarchy. In *Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 444–453. Springer, 2001.

[277] Alexandru Todor, Wojciech Lukasiewicz, Tara Athan, and Adrian Paschke. Enriching topic models with DBpedia. In *On the Move to Meaningful Internet Systems*, pages 735–751. Springer, 2016.

[278] Felix Tristram, Sebastian Walter, Philipp Cimiano, and Christina Unger. Weasel: a Machine Learning Based Approach to Entity Linking combining different features. In *NLP & DBpedia Workshop, (ISWC)*, pages 25–32. CEUR-WS.org, 2015.

[279] Christina Unger, André Freitas, and Philipp Cimiano. An introduction to question answering over linked data. In *Reasoning on the Web in the Big Data Era*, pages 100–140. Springer, 2014.

[280] Victoria S. Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *J. Web Sem.*, 4(1):14–28, 2006.

[281] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. AGDISTIS – Graph-based disambiguation of Named Entities using Linked Data. In *International Semantic Web Conference (ISWC)*, pages 457–471. Springer, 2014.

[282] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. Gerbil: General entity annotator benchmarking framework. In *World Wide Web Conference (WWW)*, pages 1133–1143, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[283] Jason Utt and Sebastian Padó. Ontology-based distinction between polysemy and homonymy. In *International Conference on Computational Semantics, IWCS*. The Association for Computer Linguistics, 2011.

[284] Andrea Varga, Amparo Elizabeth Cano Basave, Matthew Rowe, Fabio Ciravegna, and Yulan He. Linked knowledge sources for topic classification of microposts: A semantic graph-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 26:36–57, 2014.

[285] Paola Velardi, Paolo Fabriani, and Michele Missikoff. Using text processing techniques to automatically enrich a domain ontology. In *FOIS*, pages 270–284, 2001.

[286] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering Semantics of Tables on the Web. *PVLDB*, 4(9):528–538, 2011.

[287] Juan Antonio Lossio Ventura, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. Biomedical term extraction: overview and a new methodology. *Inf. Retr. Journal*, 19(1-2):59–99, 2016.

[288] Roberto De Virgilio. Rdfa based annotation of web pages through keyphrases extraction. In *On the Move (OTM)*, pages 644–661. Springer, 2011.

[289] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

[290] Jörg Waitelonis and Harald Sack. Augmenting video search with linked open data. In *International Conference on Semantic Systems (I-Semantics)*, pages 550–558. Verlag der Technischen Universität Graz, 2009.

[291] Christian Wartena and Rogier Brussee. Topic detection by clustering keywords. In *International Workshop on Database and Expert Systems Applications (DEXA)*, pages 54–58. IEEE, 2008.

[292] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1366–1371. ACL, 2013.

[293] Daya C. Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *J. Information Science*, 36(3):306–323, 2010.

[294] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. KEA: practical automatic keyphrase extraction. In *Conference on Digital Libraries (DL)*, pages 254–255. ACM, 1999.

[295] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):20, 2012.

[296] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question answering on freebase via relation extraction and textual evidence. *arXiv*, 2016.

[297] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *International Conference on Management of Data (SIGMOD)*, pages 97–108. ACM, 2012.

[298] Hiroyasu Yamada and Yuji Matsumoto. Statistical Dependency Analysis with Support Vector Machines. In *International Workshop on Parsing Technologies (IWPT)*, volume 3, pages 195–206, 2003.

[299] Surender Reddy Yerva, Michele Catasta, Gianluca Demartini, and Karl Aberer. Entity disambiguation in tweets leveraging user social profiles. In *Information Reuse & Integration, IRI*, pages 120–128. IEEE Computer Society, 2013.

[300] Mohamed Amir Yosef, Johannes Hoffart, Yusra Ibrahim, Artem Boldyrev, and Gerhard Weikum. Adapting AIDA for tweets. In *World Wide Web Conference (WWW)*, pages 68–69.

CEUR-WS.org, 2014.

[301] Daniel H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.

[302] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1753–1762. ACL, 2015.

[303] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. Using ontology to improve precision of terminology extraction from documents. *Expert Systems with Applications*, 36(5):9333 – 9339, 2009.

[304] Ziqi Zhang. *Named entity recognition: challenges in document annotation, gazetteer construction and disambiguation*. PhD thesis, The University of Sheffield, 2013.

[305] Ziqi Zhang. Effective and efficient semantic table interpretation using tableminer$^+$. *Semantic Web*, 8(6):921–957, 2017.

[306] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In *International Conferences on Web Intelligence, WI*, pages 82–89. IEEE Computer Society, 2012.

[307] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and Accurate Shift-Reduce Constituent Parsing. In *Annual meeting of the Association for Computational Linguistics (ACL)*, pages 434–443, 2013.

[308] Muhua Zhu, Jingbo Zhu, and Huizhen Wang. Improving shift-reduce constituency parsing with large-scale unlabeled data. *Natural Language Engineering*, 21(1):113–138, 2015.

[309] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over RDF: a graph data driven approach. In *International Conference on Management of Data (SIGMOD)*, pages 313–324. ACM, 2014.

[310] Zhe Zuo, Gjergji Kasneci, Toni Grütze, and Felix Naumann. BEL: bagging for entity linking. In *International Conference on Computational Linguistics (COLING)*, pages 2075–2086. ACL, 2014.

[311] Stefan Zwicklbauer, Christoph Einsiedler, Michael Granitzer, and Christin Seifert. Towards Disambiguating Web Tables. In *ISWC-PD*, pages 205–208. CEUR-WS.org, 2013.

[312] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Doser - A knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In *Extended Semantic Web Conference (ESWC)*, pages 182–198. Springer, 2016.

Appendix

A. Primer: Traditional Information Extraction

Information Extraction (IE) refers to the automatic extraction of implicit information from unstructured or semi-structured data sources. Along these lines, IE methods are used to identify entities, concepts and/or semantic relations that are not otherwise explicitly structured in a given source. IE is not a new area and dates back to the origins of Natural Language Processing (NLP), where it was seen as a use-case of NLP: to extract (semi-)structured data from text. Applications of IE have broadened in recent years, particularly in the context of the Web, including the areas of Knowledge Discovery, Information Retrieval, etc.

To keep this survey self-contained, in this section, we will offer a general introduction to traditional IE techniques as applied to primarily textual sources. Techniques can vary widely depending on the type of source considered (short strings, documents, forms, etc.), the available reference information considered (databases, labeled data, tags, etc.), expected results, and so forth. Rather than cover the full diversity of methods that can be found in the literature – for which we rather refer the reader to a dedicated survey such as that provided by Sarawagi [255] – our goal will be to cover core tasks and concepts found in traditional IE pipelines, as are often (re)used by works in the context of the Semantic Web. We will also focus primarily on English-centric examples and tools, though much of the discussion generalizes (assuming the availability of appropriate resources) to other languages, which we discuss as appropriate.

A.1. Core Preprocessing/NLP/IE Tasks

We begin our discussion by introducing the main tasks found in an IE pipeline considering textual input. Our discussion follows the high-level process illustrated in Figure 1 (similar overviews have been presented elsewhere; see, e.g., Bird *et al.* [20]). We assume that data have already been collected. The first task then involves cleaning and parsing data, e.g., to extract text from semi-structured sources. Thereafter, a sequence of core NLP tasks are applied to tokenize text; to find sentence boundaries; to annotate tokens with parts-of-speech tags such as nouns, determiners, etc.; and to apply parsing to group and extract a structure from the grammatical connections between individual tagged tokens. Thereafter, some initial IE tasks can be applied, such as to extract topical keywords, or identify named entities in a text, or to extract relations between entities mentioned in the text.

EXTRACTION & CLEANSING [PREPROCESSING]:

Across all of the various IE applications that may be considered, source data may come from diverse formats, such as plain text files, formatted text documents (e.g., Word, PDF, PPT), or documents with markup (e.g., HTML pages). Likewise, different formats may

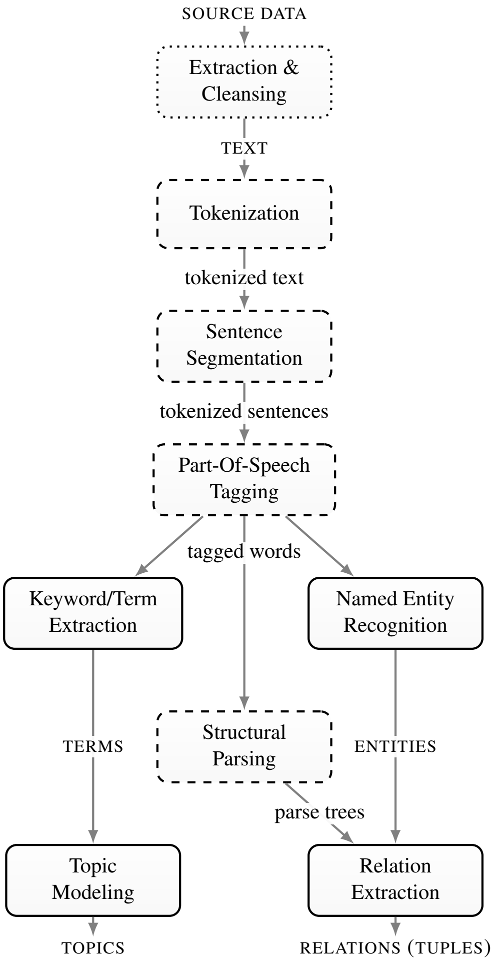

Fig. 1. Overview of a *typical* Information Extraction process, where dotted nodes indicate a preprocessing task, dashed nodes indicate a core NLP task and solid nodes indicate a core IE task; small-caps indicate potential inputs and outputs for the IE process; it is worth noting, however, that this is for the purposes of illustration: many IE process may follow a different flow to that presented here (e.g., some Keyphrase Extraction processes do not require Part-Of-Speech Tagging).

have different types of escape characters, where for example in HTML, "`&eacute;`" escapes the symbol "é". Hence, an initial pre-processing step is required to extract plain text from diverse sources and to clean that text in preparation for subsequent processing. This step thus involves, for example, recognizing and extracting text from graphics or scanned documents, stripping the sources of control strings and presentational tags, unescaping special characters, and so forth.

Example: In Listing 5, we provide an example preprocessing step where HTML markup is removed from the text and special characters are unescaped.

Listing 5: Cleansing & parsing example

```
Input:  <p><b>Bryan Lee Cranston</b> is an American
     ↪  actor. He is known for portraying &quot;
     ↪ Walter White&quot; in the drama series
     ↪ Breaking Bad.</p>
Output: Bryan Lee Cranston is an American actor.
     ↪ He is known for portraying "Walter White" in
     ↪  the drama series Breaking Bad.
```

Techniques: The techniques used for extraction and cleansing are as varied as the types of input sources that one can consider. However, we can mention that there are various tools that help extract and clean text from diverse types of sources. For HTML pages, there are various parsers and cleaners, such as the Jericho Parser[73] and HTML Tidy[74]. Other software tools help to extract text from diverse document formats – such as presentations, spreadsheets, PDF files – with a prominent example being Apache Tika[75]. For text embedded in images, scanned documents, etc., a variety of Optical Character Recognition (OCR) tools are available including, e.g., FreeOCR[76].

Text Tokenization [NLP]:

Once plain text has been extracted, the first step towards extracting further information is to apply Text Tokenization (or simply Tokenization), where text is broken down into a sequence of atomic tokens encoding words, phrases, punctuation, etc. This process is employed to identify linguistic units such as words, punctuations, numbers, etc. and in turn, to leave intact indivisible or compound words. The result is a sequence of tokens that preserves the order in which they are extracted from the input.

Example: As depicted in Listing 6 for the running example, the output of a tokenization process is composed of tokens (including words and non-whitespace punctuation marks) separated by spaces.

Listing 6: Text tokenization example

[73] http://jericho.htmlparser.net/docs/index.html
[74] http://tidy.sourceforge.net/
[75] https://tika.apache.org/
[76] http://www.free-ocr.com/

```
Input: Bryan Lee Cranston is an American actor.  He
     ↪  is known for portraying "Walter White" in
     ↪ the drama series Breaking Bad.
Output: Bryan␣Lee␣Cranston␣is␣an␣American␣actor␣.␣He
     ↪ ␣is␣known␣for␣portraying␣"␣Walter␣White␣"␣in
     ↪ ␣the␣drama␣series␣Breaking␣Bad␣.
```

Techniques: Segmentation of text into tokens is usually carried out by taking into account white spaces and punctuation characters. However, some other considerations such as abbreviations and hyphenated words[77] must be covered.

SENTENCE SEGMENTATION [NLP]:
The goal of sentence segmentation (aka sentence breaking, sentence boundary detection) is to analyze the beginnings and endings of sentences in a text. Sentence segmentation organizes a text into sequences of small, independent, grammatically self-contained clauses in preparation for subsequent processing.

Example: An example segmentation of sentences is presented in Listing 7 where sentences are output as an ordered sequence that follows the input order.

Listing 7: Sentence detection example

```
Input: Bryan Lee Cranston is an American actor . He
     ↪  is known for portraying " Walter White " in
     ↪  the drama series Breaking Bad .
Output:
1.— Bryan Lee Cranston is an American actor .
2.— He is known for portraying " Walter White " in
     ↪ the drama series Breaking Bad .
```

Techniques: Sentence boundaries are initially obtained by simply analyzing punctuation marks. However, there may be some ambiguity or noise problems when only considering punctuation (e.g., abbreviations, acronyms, misplaced characters) that affect the precision of techniques. This can be alleviated by means of lexical databases or empirical patterns (e.g., to state that the period in "`Dr.`" or the periods in "`e.g.`" will rarely denote sentence breaks, or to look at the capitalization of the subsequent token, etc.).

PART-OF-SPEECH TAGGING [NLP]:
A Part-Of-Speech (aka. POS) tagger assigns a grammatical category to each word in a given sentence, identifying verbs, nouns, adjectives, etc. The grammatical category assigned to a word often depends not only on the meaning of the word, but also its context. For example, in the context "`they took the lead`", the word "`lead`" is a noun, whereas in the context "`they lead the race`" the word "`lead`" is a verb.

Table 9
OpenNLP POS codes

POS Code	Meaning
DT:	Determiner
JJ:	Adjective
IN:	Preposition or subordinating conjunction
NN:	Noun: singular or mass
NNP:	Noun: proper, singular
PRP:	Personal pronoun
VBG:	Verb: gerund or past participle
VBZ:	Verb: 3rd person, singular, present

Example: An example POS-tagged output from the OpenNLP[78] tool is provided in Listing 8, where next to every word its grammatical role is annotated. We provide the meaning of the POS-tag codes used in this example in Table 9 (note that this is a subset of the 30+ codes supported by OpenNLP).

Listing 8: POS tagger example

```
Input:
1.— Bryan Lee Cranston is an American actor .
2.— He is known for portraying " Walter White " in
     ↪ the drama series Breaking Bad .
Output:
1. Bryan_NNP Lee_NNP Cranston_NNP is_VBZ an_DT
     ↪ American_JJ actor_NN ._.
2. He_PRP is_VBZ known_VBN for_IN portraying_VBG
     ↪ Walter_NNP White_NNP in_IN the_DT drama_NN
     ↪ series_NN Breaking_VBG Bad_JJ ._.
```

Techniques: A wide variety of techniques have been explored for POS tagging. POS taggers typically assume access to a dictionary in the given language that provides a list of the possible tags for a word; for example "`lead`" can be a verb, or a noun, or an adjective, but not a preposition or determiner. To decide between the possible tags in a given context, POS taggers may use a variety of approaches:

- Rule-based approaches use hand-crafted rules for tagging or correcting tags; for example, a rule for English may state that if the word "`to`" is followed by "`the`", then "`to`" should be tagged as a

[77] The Art of Tokenization `https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en`

[78] `https://opennlp.apache.org/`

preposition (“`to go to the bar`”), not as part of an infinitive (“`to go to the bar`”) [27,145].

- Supervised stochastic approaches learn statistics and patterns from a corpus of tagged text – for example, to indicate that “`to`” is followed by a verb *x*% of the time, by a determiner *y*% of the time, etc. – which can then be used to decide upon a most probable tag for a given context in unseen text. A variety of models can be used for learning and prediction, including Markov Models [55], Maximum Entropy [241], etc. Various corpora are available in a variety of languages with manually-labeled POS tags that can be leveraged for learning; for example, one of the most popular such corpora for English is the Penn Treebank [176].
- Unsupervised approaches do not assume a corpus nor a dictionary, but instead try to identify terms that are used frequently in similar contexts [45]. For example, determiners will often appear in similar contexts, a verb base form will often follow the term `will` in English, etc.; such signals help group words into clusters that can then be mapped to grammatical roles.
- Of course, hybrid approaches can be used, for example, to learn rules, or to perform an unsupervised clustering and then apply a supervised mapping of clusters to grammatical roles, etc.

Discussion: While POS-tagging can disambiguate the grammatical role of words, it does not tackle the problem of disambiguating words that may have multiple senses within the same grammatical role. For example, the verb “`lead`” invokes various related senses: to be in first place in a race; to be in command of an organization; to cause; etc. Hence sometimes *Word Sense Disambiguation* (WSD) is further applied to distinguish the semantic sense in which a word is used, typically with respect to a resource listing the possible senses of words under various grammatical roles, such as WordNet [195]; for further details on WSD techniques, we refer to the survey by Navigli [211].

STRUCTURAL PARSING (*Constituency*) [NLP]:
Constituency parsers are used to represent the syntactic structure of a piece of text by grouping words into phrases with specific grammatical roles, such as noun phrases, prepositional phrases, and verb phrases. More specifically, the constituency parser organizes adjacent elements of a text into groups or phrases using a context-free grammar. The output of a constituency parser is an ordered syntactic tree that denotes a hierarchical structure extracted from the text where non-terminal nodes are (increasingly high-level) types of phrases, and terminal nodes are individual words (reflecting the input order).

Table 10
OpenNLP constituency parser codes

DP Code	Meaning
NP:	Noun phrase
VP:	Verb phrase
ROOT:	Text
S:	Sentence

Example: An example of a constituency parser output given by the OpenNLP tool is shown in Listing 9, with the corresponding syntactic tree drawn in Figure 2. Some of the POS codes from Table 9 are seen again, where new higher-level constituency codes are enumerated in Table 10.

Listing 9: Constituency parser example

```
Input:  Bryan Lee Cranston is an American actor.
Output: (ROOT (S (NP (NNP Bryan) (NNP Lee) (NNP
    ↪ Cranston)) (VP (VBZ is) (NP (DT an) (JJ
    ↪ American) (NN actor))) (. .)))
```

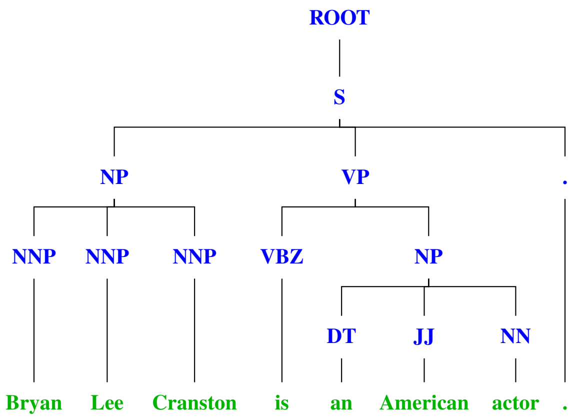

Fig. 2. Constituency parse tree example from Listing 9

Discussion: In certain scenarios, the higher-level relations in the tree may not be required. As a common example, an IE pipeline focuses on extracting information about entities may only require identification of noun phrases (NP) and not verb phrases (VP). In such scenarios, *shallow parsing* (aka. *chunking*) may be applied to construct only the lower levels of the parse tree, as needed. Conversely, *deep parsing* refers to deriving a parse-tree reflecting the entire structure of the input sentence (i.e., up to the root).

Techniques: Conceptually, structural parsing is similar to a higher-level recursion on the POS-tagging task. One of the most commonly employed techniques is to use a Probabilistic Context Free Grammar (PCFG), where terminals are associated with input probabilities from POS tagging and, thereafter, the probabilities of candidate non-terminals are given as the products of their children's probabilities multiplied by the probability of being formed from such children;[79] the probability of a candidate parse-tree is then the probability of the root of the tree, where the goal is then to find the candidate parse tree(s) that maximize(s) probability.

There are then a variety of long-established parsing methods to find the "best" constituency parse-tree based on (P)CFGs, including the Cocke–Younger–Kasami (CYK) algorithm [301], which searches for the maximal-probability parse-tree from a PCFG in a bottom-up manner using dynamic programming (aka. charting) methods. Though accurate, older methods tend to have limitations in terms of performance. Hence novel parsing approaches continue to be developed, including, for example, deterministic parsers that trade some accuracy for large gains in performance by trying to construct a single parse-tree directly; e.g., Shift–Reduce constituency parsers [307].

In terms of assigning probabilities to ultimately score various parse-tree candidates, again a wide variety of corpora (called *treebanks*) are available, offering constituency-style parse trees for real-world texts in various languages. In the case of English, for example, the Penn Treebank [176] and the British Component of the International Corpus of English (ICE-GB) [215] are frequently used for training models.

Structural Parsing (*Dependency*) [NLP]:

Dependency parsers are sometimes used as an alternative – or to complement – constituency parsers. Again, the output of a dependency parser will be an ordered tree denoting the structure of a sentence. However, the dependency parse tree does not use hierarchical nodes to group words and phrases, but rather builds a tree in a bottom-up fashion from relations between words called *dependencies*, where verbs are the dominant words in a sentence (*governors*) and other terms are recursively dependents or inferiors (*subordinates*). Thus a verb may have its subject and object as its dependents, while the subject noun in turn may have an adjective or determiner as a dependent.

Example: We provide a parse tree in Figure 3 for the running example. In the dependency parse-tree, children are direct dependents of parents. Notably, the extracted structure has some correspondence with that of the constituency parse tree, where for example we see a similar grouping on noun phrases in the hierarchy, even though noun phrases (NP) are not directly named.

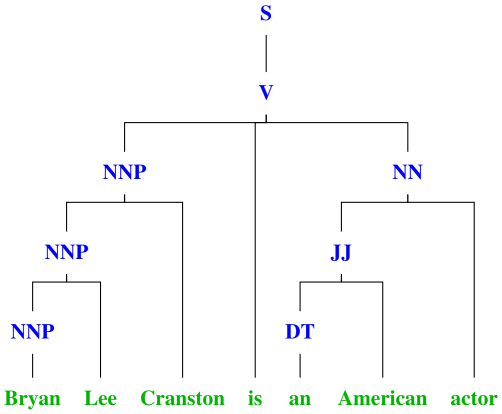

Fig. 3. Dependency parse tree example

Discussion: The choice of parse tree (constituency vs. dependency) may depend on the application: while a dependency tree is more concise and allows for quickly resolving relations, a task that requires phrases (such as NP or VP phrases) as input in a subsequent step may prefer constituency parsing methods. Hence, tools are widely available for both forms of parsing. While dependency parse-trees cannot be directly converted to constituency parse-trees, in the inverse direction, "deep" constituency parse-trees can be converted to dependency parse-trees in a deterministic manner.

Techniques: Similar techniques as discussed for constituency parsing can be applied for dependency parsing. However, techniques may naturally be better suited or may be adapted in particular ways to offer better results for one type of parsing or the other, and indeed a wide variety of dependency-specific parsers have been proposed down through the years [221], with older approaches based on dynamic program-

[79] For example, the probability of the NP "`an American actor`" in the running example would be computed as the probability of an NP being derived from (DT,JJ,NN) times the probability of "`an`" being DT, times "`American`" being JJ, times "`actor`" being NN.

ming [118,85], and newer approaches based on deterministic parsing – meaning that an approximation of the best parse-tree is constructed in “one shot” – using, for example, Support Vector Machines [298] or Neural Networks [41], and so forth.

A variety of treebanks are available for learning-based approaches with parse-trees specified per the dependency paradigm. For example, Universal Dependencies [222] offers treebanks in a variety of languages. Likewise, conversion tools exist to map constituency-based treebanks (such as the Penn Treebank [176]) to dependency-based corpora.

Named Entity Recognition [NLP/IE]:

Named Entity Recognition (NER) refers to the identification of strings in a text that refer to various types of entities. Since the 6th Message Understanding Conference (MUC-6), entity types such as Person, Organization, and Location have been accepted as standard classes by the community. However, other types like Date, Time, Percent, Money, Products, or Miscellaneous are also often used to complement the standard types recognized by MUC-6.

Example: An example of an NER task is provided in Listing 10. The output follows the accepted MUC XML format where "ENAMEX" tags are used for names, "NUMEX" tags are used for numerical entities, and "TIMEX" tags are used for temporal entities (dates). We see that Bryan Lee Cranston is identified to be a name for an entity of type Person.

Listing 10: NER example

```
Input: Bryan Lee Cranston is an American actor
Output: <ENAMEX TYPE="PERSON">Bryan Lee Cranston</
    ↪ ENAMEX> is an American actor
```

Techniques: NER can be seen as a classification problem, where given a labeled dataset, an NER algorithm assigns a category to an entity. To achieve this classification, various features in the text can be taken into account, including features relating to orthography (capitalization, punctuation, etc.), grammar (morphology, POS tags, chunks, etc.), matches to lists of words (dictionaries, stopwords, correlated words, etc.) and context (co-occurrence of words, position). Combining these features leads to more accurate classification; per Listing 11, an ambiguous name such as “Green Mile” could (based on a dictionary, for example) refer to a movie or a location, where the context is important to disambiguate the correct entity type.

Listing 11: NER context example

```
Input: The Green Mile is a prison
Output:  The <ENAMEX TYPE="LOCATION">Green Mile</
    ↪ ENAMEX> is a prison
```

Based on the aforementioned features – and following a similar theme to previous tasks – Nadeau [207] and Zhang [304] identify two kinds of approaches for performing classification: rule based and machine-learning based. The first uses hand-crafted rules or patterns based on regular expressions to detect entities, for example, matching the rule “*X* is located in *Y*” can detect locations from patterns such as “Miquihuana is located in Mexico”. However, producing and maintaining rules or patterns that describe entities from a domain in a given language is time consuming and, as a result, such rules or patterns will often be incomplete.

For that reason, many recent approaches prefer to use machine-learning techniques. Some of these techniques are supervised, learning patterns from labeled data using Hidden Markov Models, Support Vector Machines, Conditional Random Fields, Naive Bayes, etc. Other such techniques are unsupervised, and rely on clustering entity types that appear in similar contexts or appear frequently in similar documents, and so forth. Further techniques are semi-supervised, using a small labeled dataset to “bootstrap” the induction of further patterns, which, recursively, can be used to retrain the model with novel examples while trying to minimize *semantic drift* (the reinforcement of errors caused by learning from learned examples).

Term Extraction [IE]:

The goal of term extraction is to identify domain-specific phrases representing concepts and relationships that constitute the particular nomenclature of that domain. Applications of term extraction include the production of domain-specific glossaries and manuals, machine translation of domain-specific text, domain-specific modeling tasks such as taxonomy- or ontology-engineering, and so forth.

Example: An example of term extraction is depicted in Listing 12. The input text is taken from the abstract of the Wikipedia page *Breaking Bad*[80] and the output consists of the terms extracted by the TerMine service,[81] which combines linguistic and statistical analyses and returns a list of terms with a score representing

[80] https://en.wikipedia.org/wiki/Breaking_Bad

[81] http://www.nactem.ac.uk/software/termine/#form

an occurrence measure (termhood). Note that terms are generally composed of more than one word and those with same score are considered as tied.

Listing 12: Term extraction example

```
Input: Breaking Bad is an American crime drama
     ↪ television series created and produced by
     ↪ Vince Gilligan. The show originally aired on
     ↪  the AMC network for five seasons, from...
Output:
1       primetime emmy award    4.754888
2       drama series    3
2       breaking bad    3
4       american crime drama television series
     ↪ 2.321928
5       aaron paul      2
5       anna gunn       2
5       television critics association awards   2
5       drug kingpin gus fring  2
9       outstanding lead actor  1.584962
9       character todd alquist  1.584962
9       lawyer saul goodman     1.584962
9       primetime emmy awards   1.584962
9       golden globe awards     1.584962
9       fixer mike ehrmantraut  1.584962
9       guinness world records  1.584962
9       outstanding supporting actress  1.584962
9       outstanding supporting actor    1.584962
9       student jesse pinkman   1.584962
9       inoperable lung cancer  1.584962
9       elanor anne wenrich     1.584962
9       drug enforcement administration 1.584962
9       sister marie schrader   1.584962
```

Techniques: According to da Silva Conrado *et al.* [66] and Pazienza *et al.* [228], approaches for term extraction can be classified as being statistical, linguistic, or a hybrid of both. Statistical approaches typically estimate two types of measures: (i) *unithood* quantifies the extent to which multiple words naturally form a single complex term (using measures such as log likelihood, Pointwise Mutual Information, etc.) , while (ii) *termhood* refers to the strength of the relation of a term to a domain with respect to its specificity for that domain or the domain's dependence on that term (measured using TF–IDF, etc.). Linguistic approaches rather rely on a prior NLP analysis to identify POS tags, chunks, syntactic trees, etc., in order to thereafter detect syntactic patterns that are characteristic for the domain. Often statistical and linguistic approaches are combined, as per the popular *C-value* and *NC-value* measures [97], which were used by TerMine to generate the aforementioned example.

KEYPHRASE EXTRACTION [IE]:
Keyphrase extraction (aka. keyword extraction) refers to the task of identifying keyphrases (potentially multi-word phrases) that characterize the subject of a document. Typically such keyphrases are noun phrases that appear frequently in a given document relative to other documents in a given corpus and are thus deemed to characterize that document. Thus while term extraction focuses on extracting keyphrases that describe a domain, keyphrase extracts phrases that describe a document. Keyphrase extraction has a wide variety of applications, particularly in the area of Information Retrieval, where keyphrases can be used to summarize documents, or to organize documents based on a taxonomy or tagging system that users can leverage to refine their search needs. The precise nature of desired keyphrases may be highly application-sensitive: for example, in some applications, concrete entities (e.g., Bryan Lee Cranston) may be more desirable than abstract concepts (e.g., actor), while in other applications, the inverse may be true. Relatedly, there are two general settings under which keyphrase extraction can be performed [262]: *assignment* assumes a given set of keywords that are subsequently assigned to input documents, whereas *extraction* resolves keywords from the documents themselves.

Example: An example of keyphrase extraction is presented in Listing 13. Here the input is the same as in the term extraction example, and the output was obtained with a python implementation of the RAKE algorithm[82] [252] which returns ranked terms favored by a word co-occurrence based score.

Listing 13: Keyphrase extraction example

```
Input: Breaking Bad is an American crime drama
     ↪ television series created and produced by
     ↪ Vince Gilligan. The show originally aired on
     ↪  the AMC network for five seasons, from...
Output (subset):  [('struggling high school
     ↪ chemistry teacher diagnosed', 36.0), ('
     ↪ american crime drama television series
     ↪ created', 33.5), ('selling crystallized
     ↪ methamphetamine', 9.0), ('southern
     ↪ colloquialism meaning', 9.0), ('student
     ↪ jesse pinkman', 9.0), ('show originally
     ↪ aired', 9.0), ('inoperable lung cancer',
     ↪ 9.0), ('amc network', 4.0), ('criminal world
     ↪ ', 4.0), ('aaron paul', 4.0), ('raise hell',
     ↪  4.0), ('vince gilligan', 4.0)]
```

Techniques: Medelyan [183] identifies three conceptual steps commonly found in keyphrase extraction algorithms: candidate selection, property calculation, and scoring and selection. Candidate selection extracts words or phrases that can potentially be keywords,

[82] `https://github.com/aneesha/RAKE`

where a variety of linguistic, rule-based, statistical and machine-learning approaches can be employed. Property calculation associates candidates with features extracted from the text, which may include measures such as Mutual Information, TF–IDF scoring, placement of words in the document (title, abstract, etc.), and so forth. Scoring and selection then uses these features to rank candidates and select a final set of keywords, using, e.g., direct calculations with heuristic formulae, machine-learning methods, etc.

Topic Modeling [IE]:

In the context of topic modeling [21], a topic is a cluster of keywords that is viewed intuitively as representing a latent semantic theme present in a document; such topics are computed based on probability distributions over terms in a text. Thus while keyphrase extraction is a syntactic process that results in a flat set of keywords that characterize a given document, topic modeling is a semantic process that applies a higher level clustering of such keywords based on statistical models that capture the likelihood of semantically-related terms appearing in a given context (e.g., to capture the idea that "`boat`" and "`wave`" relate to the same abstract theme, whereas "`light`" and "`wave`" relate to a different such theme).

Example: We present in Listing 14 an example of topics extracted from the text of 43 Wikipedia pages linked from Bryan Cranston's filmography[83]; we input multiple documents to provide enough text to derive meaningful topics. The topics were obtained using the Mallet tool[84], which implements the LDA algorithm (set to generate 20 topics with 10 keywords each). Each line lists a topic, with a topic ID, a weight, and a list of the top-ranked keywords that form the topic. The two most highly-weighted topics show keywords such as *movies, production, cast, character*, and *cranston* which capture a general overview of the input dataset. Other topics contain keywords pertaining to particular movies, such as *gordon* and *batman*.

Listing 14: Topic Modeling example

```
Input: [Bryan Cranston filmography Wikipedia pages]
Output:

7       0.43653 film cast released based release
    ↪ production cranston march office bryan
9       0.39279 film's time original made films
    ↪ character set movie soundtrack box
6       0.35986 back people tells home find day man
    ↪  son car night
10      0.31798 gordon batman house howard story
    ↪ police alice job donald wife
3       0.31769 haller cut reviews chicago working
    ↪ hired death agent international martinez
12      0.28737 characters project story dvd space
    ↪ production members force footage called
13      0.26046 producers miss awards sunshine
    ↪ academy family festival script richard
    ↪ filming
8       0.23662 million kung opening animated panda
    ↪  weekend release highest-grossing animation
    ↪ china
19      0.23053 war ryan american world television
    ↪ beach miller private spielberg saving
0       0.218   voice canadian moon british cia
    ↪ argo iran historical u.s tehran
18      0.21697 red tuskegee quaid airmen tails
    ↪ story lucas total easy recall
11      0.21486 hanks larry tom band thing mercedes
    ↪  song wonders guy faye
14      0.18696 drive driver refn trumbo festival
    ↪ september franco gosling international irene
2       0.18516 rock sherrie julianne cruise hough
    ↪ drew tom chloe diego stacee
5       0.1823  laird rangers ned madagascar power
    ↪ circus released stephanie alex company
17      0.17892 version macross released fighter
    ↪ english japanese movie street release series
16      0.17171 contagion soderbergh virus burns
    ↪ cheever pandemic public vaccine health
    ↪ emhoff
15      0.15989 godzilla legendary edwards stating
    ↪ toho monster nuclear stated san muto
1       0.14733 carter armitage ross john mars
    ↪ disney burroughs stanton earth iii
4       0.12169 rama sita ravana battle hanuman
    ↪ king lanka lakshmana ramayana indrajit
```

Techniques: Techniques for topic modeling often rely on the *distributional hypothesis* that similar words tend to appear in similar contexts; in this case, the hypothesis for topic modeling is that words on the same "topic" will often appear grouped together in a text.

A seminal approach for modeling topics in text is *Latent Semantic Analysis* (LSA).[85] The key idea of LSA is to compute a low-dimension representation of the words in a document by "compressing" words that frequently co-occur. For this purpose, given a set of documents, LSA first computes a matrix M with words as rows and documents as columns, where value (i, j) denotes how often word w_i appears in document d_j. Often stop-word removal will be applied beforehand; however, the resulting matrix may still have very high dimensionality and it is often desirable to "compress" the rows in M by combining similar words (by virtue of co-occurring frequently, or in other words, having sim-

[83] `https://en.wikipedia.org/wiki/Bryan_Cranston_filmography`

[84] `http://mallet.cs.umass.edu/topics.php`

[85] Often interchangeably called Latent Semantic Indexing (LSI): a particular implementation of the LSA model.

ilar values in their row) into one dimension. For this purpose, LSA applies a linear algebra technique called Singular Value Decomposition (SVD) on M, resulting in a lower-dimensional bag-of-topics representation of the data; a resulting topic is then a linear combination of base words, for example, (“`tumour`” $\times$ 0.4 + “`malignant`” $\times$ 1.2 + “`chemo`” $\times$ 0.8), here compressing three dimensions into one per the given formula. The resulting matrix can be used to compare documents (e.g., taking the dot product of their columns) with fewer dimensions versus M (and minimal error), and in so doing, the transformation simultaneously groups words that co-occur frequently into “topics”.

Rather than using linear algebra, the *probabilistic LSA* (pLSA) [131] variant instead applies a probabilistic model to compute topics. In more detail, pLSA assumes that documents are sequences of words associated with certain probabilities of being generated. However, which words are generated is assumed to be governed by a given latent (hidden) variable: a *topic*. Likewise, a document has a certain probability of being on a given topic. The resulting model thus depends on two sets of parameters: the probability of a document being on a given topic (e.g., we can imagine a topic as *cancer*, though topics are latent rather than explicitly named), and the probability of a word being used to speak about a given topic (e.g., “`tumour`” would have a higher probability of appearing in a document about cancer than “`wardrobe`”, even assuming both appear with similar frequency in general). These two sets of parameters can be learned on the basis that they should predict how words are distributed amongst the given documents – how the given documents are generated – using probabilistic inferencing methods such as Expectation–Maximization (EM).

The third popular variant of a topic model is *Latent Dirichlet Allocation* (LDA) [22], which, like pLSA, also assumes that a document is associated with potentially multiple latent topics, and that each topic has a certain probability of generating a particular word. The main novelty of LDA versus pLSA is to assume that topics are distributed across documents, and words distributed across topics, according to a *sparse Dirichlet prior*, which are associated, respectively, with two parameters. The intuition for considering a sparse Dirichlet prior is that topics are assumed (without any evidence but rather as part of the model's design) to be strongly associated with a few words, and documents with a few topics. To learn the various topic distributions involved that can generate the observed word distributions in the given documents, again, methods such as EM or Gibb's sampling can be applied.

A number of topic modeling variants have also been proposed. For example, while the above approaches are unsupervised, supervised variants have been proposed that guide the modeling of topics based on manual input values [180,271]; for instance, supervised topic modeling can be used to determine whether or not movie reviews are positive or negative by training on labeled examples, rather than relying on an unsupervised algorithm that may unhelpfully (for that application) model the themes of the movies reviewed [180]. Clustering techniques [291] have also been proposed for the purposes of *topic identification*, where, unlike topic modeling, each topic found in a text is assigned an overall label (e.g., a topic with *boat* and *wave* may form a cluster labeled *marine*).

Coreference Resolution [NLP/IE]:

While entities can be directly named, in subsequent mentions, they can also be referred to by pronouns (e.g., “`it`”) or more general forms of noun phrase (e.g., “`this hit TV series`”). The aim of coreference resolution is to thus identify all such expressions that mention a particular entity in a given text.

Example: We provide a simple example of an input and output for the coreference resolution task. We see that the pronoun “`he`” is identified as a reference to the actor “`Bryan Lee Cranston`”.

Listing 15: Coreference resolution example

```
Input:
1.- Bryan Lee Cranston is an American actor .
2.- He is known for portraying " Walter White " in
    ↪ the drama series Breaking Bad .
Output:
1.- Bryan Lee Cranston is an American actor .
2.- He [Bryan Lee Cranston] is known for portraying
    ↪  " Walter White " in the drama series
    ↪ Breaking Bad .
```

Techniques: An intuitive approach to coreference resolution is to find the closest preceding entity that “`matches`” the referent under analysis. For example, the pronoun “he” in English should be matched to a male being and not, for example, an inanimate object or a female being. Likewise, number agreement can serve as a useful signal. The more specific the referential expression, and the more specific the information available about individual entities (as generated, for example, by a prior NER execution), the better the quality of the output of the coreference resolution phase. However, a generic pronoun such as "`it`" in English is more difficult to resolve, potentially leaving multiple ambiguous candidates to choose from. In this case, the context in which the referential expression appears is important to help resolve the correct entity.

Various approaches for coreference resolution have been introduced down through the years. Most approaches are supervised, requiring the use of labeled data and often machine learning methods, such as Decision Trees [181], Hidden Markov Models [268], or more recently, Deep Reinforcement Learning [53]. Unsupervised approaches have also been proposed, based on, for example, Markov Logic [235].

RELATION EXTRACTION [NLP/IE]:

Relation Extraction (RE) [11] is the task of identifying semantic relations from text, where a semantic relation is a tuple of *arguments* (entities, things, concepts) with a semantic fragment acting as *predicate* (noun, verb, preposition). Depending on the number of arguments, a relation may be unary (one argument), binary (two arguments), or *n*-ary ($n > 2$ arguments).

Example: In Listing 16, we show the result of an example (binary) relation extraction process. The input most closely resembles a ternary relation, with "`published in`" as predicate, and "`the results`", "`Physical Review Letters`" and "`June`" as arguments. The first element of each binary relation is called the *subject*, the second is called the *relation phrase* or *predicate*, and the last is called the *object*. As illustrated in the example, coreference resolution may be an important initial step to the relation extraction process to avoid having generic referents such as "`the results`" appearing in the extracted relations.

Listing 16: Relation Extraction example

```
Input: The results were published in Physical
    ↪ Review Letters in June.
Output:          [The results, were published, in
    ↪ Physical Review Letters]
[The results, were published, in June]
```

Techniques: A typical RE process typically applies a number of preliminary steps, most often to generate a dependency parse tree; further steps, such as coreference resolution, may also be applied. Following previous discussion, the RE process can then follow one of three strategies, as outlined by Banko *et al.* [14]: knowledge-based methods, supervised methods, and self-supervised methods. Knowledge-based methods are those that rely on manually-specified pattern-matching rules; supervised methods require a training dataset with labeled examples of sentences containing positive and negative relations; self-supervised methods learn to label their own training data sets.

As a part of self-supervised methods, Banko *et al.* [14] – in their TextRunner system – proposed the idea of *Open Information Extraction* (OpenIE), whose main goal is to extract relations with no restriction about a specific domain. OpenIE has attracted a lot of recent attention, where diverse approaches and implementations have been proposed. Such approaches mostly use pattern matching and/or labeled data to bootstrap an iterative learning process that broadens or specializes the relations recognized. Some of the most notable initiatives in this area are ClausIE [59], which uses a dependency parser to identify patterns called *clauses*[86] for bootstrapping; Never-Ending Language Learner (NELL), which involves learning various functions over features for (continuously) extracting and classifying entities and relations [201]; OpenIE [178][87], which uses previously labeled data and dependency pattern-matching for bootstrapping; ReVerb [89][88], which applies logistic regression over various syntactic and lexical features; and Stanford OIE[89], which uses pattern matching on dependency parse trees to bootstrap learning.

A.2. Resources and Tools

A number of tools are available to assist with the previously described NLP tasks, amongst the most popular of which are:

Apache OpenNLP [13] is implemented in Java and provides support for tokenization, sentence segmentation, POS tagging, constituency-based parsing, and coreference resolution. Most of the implemented methods rely on supervised learning using either Maximum Entropy or Perceptron models, where pre-built models are made available for a variety of the most widely-spoken languages; for example, pre-built English models are trained from the Penn Treebank.

[86] Clauses are fragments of a sentence that express coherent ideas and contain a subject, verb, and optionally objects.

[87] `http://openie.allenai.org/`

[88] `http://reverb.cs.washington.edu/`

[89] `http://stanfordnlp.github.io/CoreNLP/openie.html`

GATE [64] (General Architecture for Text Engineering) offers Java libraries for tokenizing text, sentence segmentation, POS tagging, parsing, and coreference resolution. The POS tagger is rule-based (a Brill parser [27]) [122] while a variety of plugins are available for integrating various alternative parsing algorithms.

LingPipe [35] is implemented in Java and offers support for tokenization, sentence segmentation, POS tagging, coreference resolution, spelling correction, and classification. Tagging, entity extraction, and classification are based on Hidden Markov Models (HMM) and *n*-gram language models that define probability distributions over strings from an attached alphabet of characters.

NLTK [20] (Natural Language Toolkit) is developed in Python and supports tokenization, sentence segmentation, POS tagging, dependency-based parsing, and coreference resolution. The POS tagger uses a Maximum Entropy model (with an English model trained from the Penn Treebank). The parsing module is based on dynamic programming (chart parsing), while coreference resolution is supported through external packages.

Stanford CoreNLP [175] is implemented in Java and supports tokenization, sentence segmentation, POS tagging, constituency parsing, dependency parsing and coreference resolution. Stanford NER is based on linear chain Conditional Random Field (CRF) sequence models The POS tagger is based on a Maximum Entropy model (with an English model trained from the Penn Treebank). The recommended constituency parser is based on a shift–reduce algorithm [308], while the recommended dependency parser uses Neural Networks [41]. A variety of coreference resolution algorithms are provided. [94].

The output of these core NLP tasks is expressed by different systems in different formats (e.g., XML, JSON), following different conference specifications (e.g., MUC, CoNLL) and standards (e.g., UIMA [92]).

A.3. Summary and discussion

Many of the techniques for the various core NLP/IE tasks described in this section fall into two high-level categories: *rule-based approaches*, where experts create patterns in a given language that provide insights on individual tokens, their interrelation, and their semantics; and *learning-based approaches* that can be either *supervised*, using labeled corpora to build models; *unsupervised*, where statistical and clustering methods are applied to identify similarities in how tokens are used; and *semi-supervised*, where seed patterns or examples learned from a small labeled model are used to recursively learn further patterns.

Older approaches often relied on a more brute-force, rule-based or supervised paradigm to processing natural language. However, continual improvements in the area of Machine Learning, together with the ever increasing computational capacity of modern machines and the wide availability of diverse corpora, mean that more and more modern NLP algorithms incorporate machine-learning models for semi-supervised methods over large volumes of diverse data.