

# Named Entity Extraction for Knowledge Graphs: A Literature Overview

TAREQ AL-MOSLMI<sup>1</sup>, MARC GALLOFRÉ OCAÑA<sup>1</sup>, ANDREAS L. OPDAHL, AND CSABA VERES

Department of Information Science and Media Studies, University of Bergen, 5007 Bergen, Norway

Corresponding author: Tareq Al-Moslmi (tareq.al-moslmi@uib.no)

This work was supported by the News Angler Project through the Norwegian Research Council under Project 275872.

**ABSTRACT** An enormous amount of digital information is expressed as natural-language (NL) text that is not easily processable by computers. Knowledge Graphs (KG) offer a widely used format for representing information in computer-processable form. Natural Language Processing (NLP) is therefore needed for mining (or lifting) knowledge graphs from NL texts. A central part of the problem is to extract the named entities in the text. The paper presents an overview of recent advances in this area, covering: Named Entity Recognition (NER), Named Entity Disambiguation (NED), and Named Entity Linking (NEL). We comment that many approaches to NED and NEL are based on older approaches to NER and need to leverage the outputs of state-of-the-art NER systems. There is also a need for standard methods to evaluate and compare named-entity extraction approaches. We observe that NEL has recently moved from being stepwise and isolated into an integrated process along two dimensions: the first is that previously sequential steps are now being integrated into end-to-end processes, and the second is that entities that were previously analysed in isolation are now being lifted in each other's context. The current culmination of these trends are the deep-learning approaches that have recently reported promising results.

**INDEX TERMS** Knowledge graphs, natural-language processing, named-entity extraction, named-entity recognition, named-entity disambiguation, named-entity linking.

## I. INTRODUCTION

*Knowledge Graphs (KG)* [1]–[3] were introduced to wider use by Google in 2012 to precisely interlink data that could be used, in their case, to assist search queries [4]. In a KG, the nodes represent either concrete objects, concepts, information resources, or data about them, and the edges represent semantic relations between the nodes [5]. Knowledge graphs thus offer a widely used format for representing information in computer-processable form. They build on, and are heavily inspired by, Tim Berners-Lee's vision of the *semantic web*, a machine-processable web of data that augments the original web of human-readable documents [6]. KGs can therefore leverage existing standards such as RDF, RDFS, and OWL. In practice, however, KGs tend to be less formal than the ontology-based semantic web promoted in [6].

At the same time, an abundance of information on the internet is being expressed as natural language (NL) text that is not easily processable by computers. Making all this text available to computers requires *Natural Language*

*Processing (NLP)* and other types of information extraction techniques. One of the central challenges is to *identify the entities mentioned in the text*. These entities can be either *named entities* that refer to individuals or abstract *concepts*.

Because entities can be represented as nodes in KGs and relations as edges, KGs are a natural way of representing NL text in computer-processable form. This paper therefore presents a literature overview of recent advances in one research area that is central for lifting NL texts to KGs: that of *extracting named entities from texts*, or *Named Entity Extraction (NEE)* [7]. NEE is a task that involves *recognising* the mention of the named entity in the text (NER), *disambiguating* its possible references (NED), and *linking* the named entity to an object in a knowledge base (NEL).

In this fast-moving area, the paper presents an overview of how the research front has moved over the last 5 years. To identify suitable articles we have searched six digital libraries: ACM, IEEE, Science Direct, Springer, WoS, and Google Scholar. Papers that investigated approaches to named-entity extraction, were published in 2014–2019, and had sufficient quality were then considered further, whereas low-quality or out-of-scope papers were excluded.

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Hugo Albuquerque<sup>1</sup>.

We searched for papers that explored at least one of the main tasks (NER, NED, and NEL) on the path from NL text to KG. A total of 362 articles were identified based on abstract filtering, and 89 papers remained after closer screening. These remaining main papers were categorized as either NER, NED, or NEL core papers according to the task they focused on. We used snowballing to identify a few additional papers referenced by other main papers.

The rest of the paper is organized as follows: Section II presents the background for our work. Section III constitutes the bulk of the paper and reviews recent approaches to lifting named entities in NL texts to KGs. Section IV discusses the current state of the art, before Section V concludes the paper and suggests further work.

## II. BACKGROUND

This section explains the most central concepts used in the paper.

### A. NATURAL LANGUAGE PROCESSING

Natural-language processing (NLP) attempts to enable computers to process human language in meaningful ways [8]. For example, [9] describes NLP as “an area of research and application that explores how computers can be used to understand and manipulate natural-language text or speech to do useful things”. NLP is commonly used to derive semantics from text or speech and to encode it in a structured format that is suitable for semantic search [7] and other types of computer processing. Many well-established techniques and tools are already available for this purpose, such as GATE [10] and other NLP pipelines; IBM Watson<sup>1</sup> and other NL analysis and lifting services; NLTK [11], Stanford CoreNLP [12], DBpedia Spotlight [13], and other NL programming APIs; OpenIE<sup>2</sup>, MinIE [14], and other information extraction tools.

### B. KNOWLEDGE GRAPHS

A knowledge graph (KG) represents semantic data as triples (i.e., as ordered sets of terms) composed as  $(s, p, o)$ : subject  $s$ , predicate  $p$ , object  $o$ , that can be either IRIs (Internationalized Resource Identifier) [15]  $i \in I$ , blank nodes  $b \in B$  or literals  $l \in L$ , so that  $s \in I \cup B$ ,  $p \in I$  and  $o \in I \cup B \cup L$  [16].

The IRIs used as subjects, predicates, and objects can be taken from well-defined vocabularies or ontologies in the Linked Open Data (LOD) cloud [5], [17], [18] that attempt to define their meaning as precisely as possible. The literal values used as objects can be represented using well-defined data types, such as the ones defined by XML Schema Definition (XSD). Through the use of terms with well-defined meanings for types, instances, and relations, the knowledge graph aims to describe the semantics of real-world entities and their relations precisely and to link the descriptions to further information in semantic LOD repositories [18].



FIGURE 1. From natural language (NL) to knowledge graphs (KG).

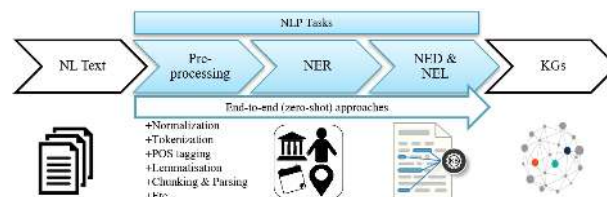


FIGURE 2. Natural-language processing (NLP) tasks.

Figure 1 shows the path from collections of NL documents through NLP to KGs.

## III. REPRESENTING NAMED ENTITIES IN KGs

A *named entity* is an individual such as a person, organization, location, or event. A *mention* is a piece of text that refers to an entity.

As already mentioned, extracting named entities from texts and representing them as nodes in KGs involves three main tasks: Named Entity Recognition (NER) attempts to find every segment in a text that mentions a named entity. Named Entity Disambiguation (NED) attempts to determine which named entity a mention refers to; for example, the mention “Trump” can refer to either a person, a corporation or a building. Named Entity Linking (NEL) attempts to provide a standard IRI for each disambiguated entity; for example, Trump-the-president can be linked to the IRI that represents him in Wikidata: <https://www.wikidata.org/entity/Q22686>. NED and NEL are closely entwined, because an ambiguous entity must be disambiguated before it can be linked and because an IRI is a good way to represent the result of disambiguation. Therefore, we will often discuss NED and NEL together.

Figure 2 shows the resulting sequence of tasks from NL to KGs. The figure also shows the more recent end-to-end (also called zero-shot) approaches, which lump together all three tasks, typically using deep neural networks. These approaches usually rely on standard pre-processing steps, which we will review first, before we go on to the other tasks.

### A. PRE-PROCESSING

The most frequently used pre-processing techniques in NL lifting are tokenization [19]–[22] and part-of-speech (POS) tagging [19], [20], [22]–[25]. Other common techniques include: stop-word removal [26], normalization [20], sentence splitting [20], [21], [25], lemmatization [19], chunking and dependency parsing [25], and structural parsing [20]. Although pre-processing is used by default, some studies do not describe in detail how they are performed. A possible reason is that most studies, especially on NER, use standard datasets that have already been pre-processed. For example,

<sup>1</sup><https://www.ibm.com/watson>

<sup>2</sup><https://nlp.stanford.edu/software/openie.html>

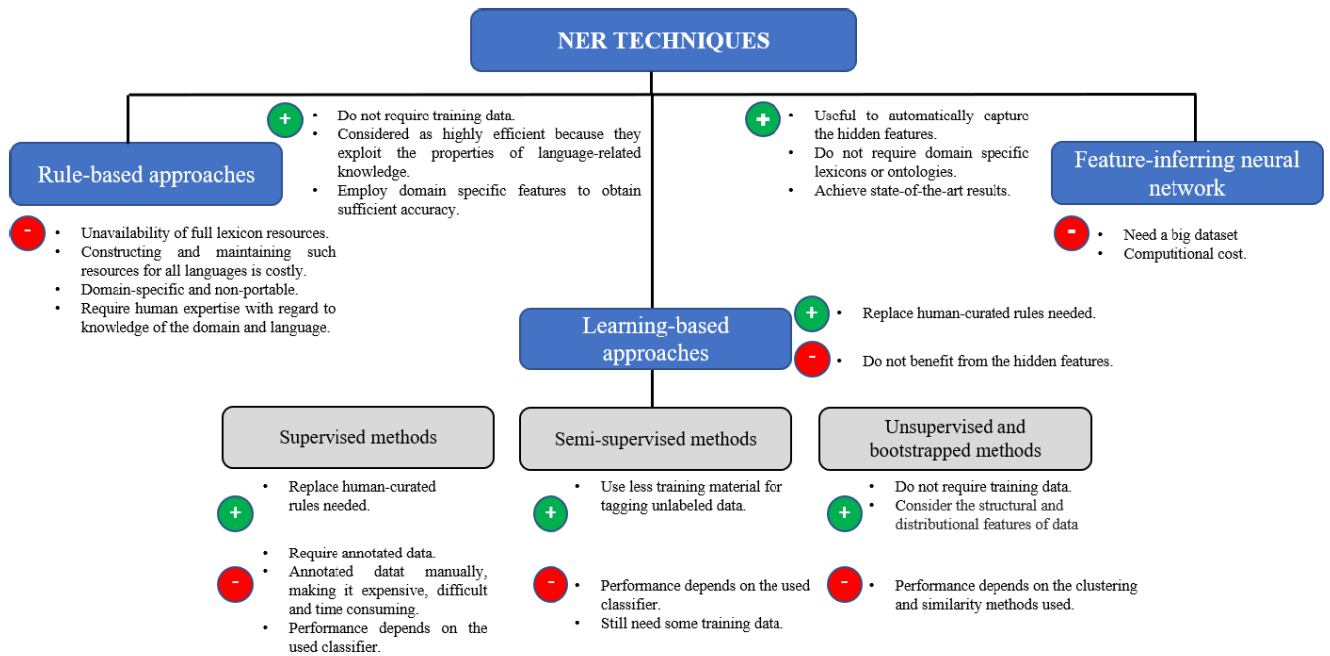


FIGURE 3. Named-entity recognition (NER) approaches.

CoNLL2003, the most popular dataset for NER, has already applied tokenization, POS tagging, and a chunking to the raw data.

The appropriate choice of pre-processing technique depends on the lifting technique to be used. For example, removing stop words might be good for a Bag-of-Words (BOW) based approach or for a model that does not consider word order, but deep-learning approaches might leverage stop words to disambiguate entities that have different meanings. Recent contributions indicate that robust NED and NEL systems require accurate tuning of several prior steps, especially tokenization and semantic similarity [27]. Recently, deep neural networks, in particular the end-to-end approaches, have reduced the need for pre-processing steps. Using deep neural networks for pre-processing tasks such as tokenization has also produced promising results [28].

Data quality plays a key role in selecting the most suitable pre-processing technique too. For example, most gold-standard datasets do not require the same pre-processing as raw-web or real-time streaming data, from which cleaning and normalization are needed to remove unnecessary or noisy terms (like emojis, currency symbols, hashtags, and so forth).

### B. NAMED ENTITY RECOGNITION

NER was first introduced in [29]. According to [30], the purpose of NER is to identify named entities contained in a text like persons, locations, organizations, time, clinical procedures, money, biological proteins, etc.

NER is a rapidly evolving research field [31]. Most of the proposed approaches have been domain-specific, limiting themselves to for example news, reviews, etc. We divide

them into three main categories following [31], as shown in Figure 3:

- The first, and earliest, category is the rule- or knowledge-based approaches [32], [33]. Most studies in this category are based on hand-crafted rules [31], [33]. The advantage of such methods is that they do not require annotated training data since they rely on lexical resources. Another advantage is that the precision of handcrafted methods can become high because of the lexicons and domain-specific knowledge. The disadvantage is that this also makes them domain dependent [30], that lexicon resources may be unavailable, and that constructing and maintaining such resources for many languages is costly [34].
- The second category of NER systems is the learning-based approaches [35]–[37]. These models are used to replace the human-curated rules needed by the first category. The methods in this category can be divided into three types: supervised, semi-supervised, and unsupervised. In supervised and semi-supervised methods, a machine-learning model is trained on input examples together with their targeted outputs. Support Vector Machines (SVM), Hidden Markov Models (HMM), Conditional Random Fields (CRF), and decision trees are common in this category [30]. NER accuracy is sometimes limited by the used classifier. For example, when HMM and SVM are employed, the dependencies among words are not considered. The unsupervised and bootstrapped methods [32], [38], [39] are more automated, although they need a minimal training dataset (seeds). Although these methods do not require as much

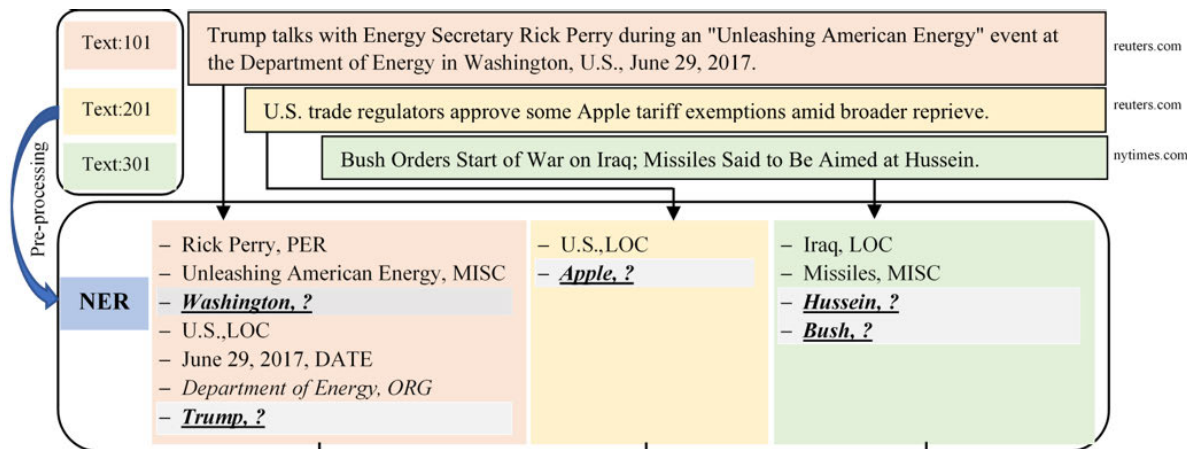


FIGURE 4. Example of a NER task.

effort as the first category, seed data is still needed for training. Moreover, they do not benefit from the feature inference of the third category below [31].

- The last, and most recent, category is the feature-inferring neural-network approaches [40]–[45]. They rely on machine learning like the previous category, but they differ from the rule- and learning-based approaches by automatically inferring features through deep learning. Recent research reports that they thus outperform earlier methods [30], [31], [46]. Unlike the above-mentioned approaches, they do not require seeds, ontologies, or domain specific lexicons and are thus more domain-independent. They also benefit from the precision of their inferred features. On the other hand, big datasets are needed to build robust models.

Although numerous NER studies have been reported recently, few have been used for semantic lifting or received attention in KG research. There may be at least two reasons. The first reason is that NER is an initial step for many other tasks — such as sentiment analysis, concept extraction, event identification and so forth — that have received more attention than semantic lifting. The second reason is that semantic lifting research has focused on the KG-construction side, leaving the NER task to off-the-shelf APIs and tools. However, using standard systems suffers from configuration restrictions and makes the combination of or switching between different NER solutions more difficult [47].

This paper will not review NER approaches in greater detail, because they are already well summarized in recent surveys [30], [31], [31]. However, to the best of our knowledge, these state-of-the-art NER methods [40]–[44] have not yet been used in pipelines that lift NL to KGs. Hence, there is a need to exploit their outputs to achieve more precise and complete lifting.

The rest of this paper will use a running example of the main lifting tasks on three input texts. We assume that the text has already been normalized and pre-processed. In the first stage, the NER task extracts all the entity candidates in the text as shown in Figure 4.

### C. NAMED ENTITY DISAMBIGUATION

Based on how they rank candidate entities, NED approaches can be classified into three categories:

#### 1) TRADITIONAL NED APPROACHES

Traditional NED studies typically use hand-designed features to calculate the similarity between the mention and its candidate entities [48], [49]. These studies can be further subdivided into independent (or individual) and collective approaches.

- Independent approaches [13], [50] use semantic similarity techniques to rank candidate entities solely according to their lexical similarity and/or empirical co-occurrence with mentions. In these methods, each mention is independently disambiguated, and they consider entity disambiguation as a ranking problem that picks the entity with the highest confidence. The confidence value is scored by combining hand-designed features extracted from the mention’s context (surrounding words) with textual descriptions of the candidate entity, for example using text about it from Wikipedia. Different similarity measures have been proposed, such as BOW-based cosine similarity. Because hand-designed features tend to contain little textual information, the accuracy of such approaches decreases in complex cases [49]. Moreover, these methods fail to catch interactions between mentions in the same document [51].
- Collective approaches [19], [52]–[54] also rely on semantic similarity, but they take into account that what is mentioned in the same (part of a) text tends to be about the same topic and that co-occurring entities should therefore often be semantically related [19], [27]. Most commonly used collective approaches thus establish the associations between candidate entities and build the mention-entity pairs using a probabilistic graph approach. AGDISTIS [52], Babelify [53] and TagME [55] use graph connectedness to exploit semantic relations between disambiguation candidates

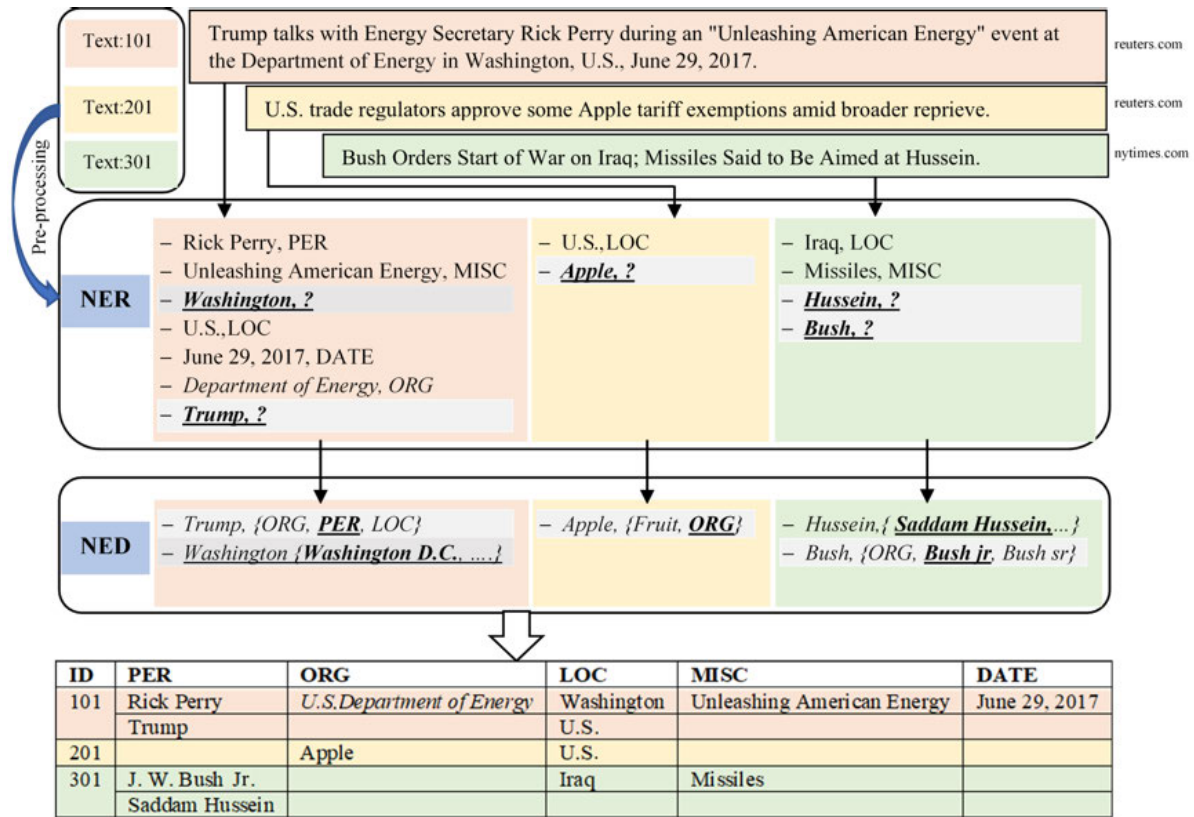


FIGURE 5. Example of a NED task.

described in a knowledge base (KB). In fact, feature representations of entities and mentions are the key factors for most of these NED approaches. Most of them rely on BOW [52], [53], which, in general, has some shortcomings, such as ignoring word meanings and expensive computation [50]. The methods thus do not exploit latent features in mention contexts and candidate descriptions [51]. A well-matched entity group can be identified using either random walks [56], Pagerank [54], [57], or dense sub-graph computations. Although the collective approaches perform more robustly than the independent ones, computation costs grow rapidly as the numbers of mentions and the lengths of documents increase. [58], [59] proposes a simplified collective pair-linking approach, which resolves the candidate entities pair-by-pair in order to decrease computational cost and complexity. In hand-crafted feature-based approaches, it is difficult for NED systems to fully leverage the inherent semantics of mention contexts and entity descriptions, which is pivotal for NED accuracy. This is due to the limitations of hand-designed features, which capture lexical information based on the surface form of the text [49].

## 2) NEURAL-NETWORK (NN) APPROACHES

In recent years, neural-network (NN) based methods [51], [58], [60], [61] have become more common and achieved

competitive results. One family of approaches map words into continuous vector spaces using word2vec or similar models [62] that comprise more semantic information than traditional BOW representations. Another family uses deep NNs to learn the latent semantic features automatically. NED accuracy is thereby enhanced along with the model's generalization ability.

The majority of the earlier NN-based approaches [63], [64] grant all the words in the mention context equal importance, which is adequate for many practical cases [49], [58], [65]. More recently, attention mechanisms [58], [65]–[67] have been introduced to assign graded importance. However as mentioned in [49], most of these methods only apply attention to mention contexts and omit the entity side. Also, they only apply attention to a single aspect of knowledge, which may not be sufficient in complex circumstances, for example with high noise contexts or less popular entities. Recently, [49] proposed a multi-perspective attention NN model to enrich mentions and entities representation in different perspectives, to capture more informative features and improve accuracy. Empirical evaluations show that NN-based approaches are effective in NED systems [49], [51], [61], [65].

In Figure 5, the running example text contains disambiguation candidates such as “Washington”, “Trump”, “Apple”, “Hussein” and “Bush”, calling for NED to be applied. The task becomes more challenging because mentions must often

be classified on several levels. An example is “Bush” where the NED system must both recognize whether it refers to a person or organization and, in the first case, recognize whether it refers to Bush Jr. or Sr. Another challenging mention is “Hussein”, which the NED system has to recognize in a context that mentions “IRAQ”, “Bush Sr.”, and “Missiles”.

The accuracy of existing NED systems is still far from perfect, especially when dealing with short texts that have little context [19], [27]. Pangloss [27] is a system for entity disambiguation and linking of noisy text. It uses a semantic similarity engine that depends on context-dependent document embeddings. It also leverages a local database to store its metadata and statistics, which enables fast disambiguation when processing streams and on low-memory devices such as mobile phones.

Few papers have employed KGs for NED tasks [19], [68]. [19] proposes to exploit existing KGs with associated texts based on unsupervised semantic similarity. The study has compared graph-based and corpus-based approaches to unsupervised similarity-based NED on real-world datasets. The study uses WordNet [69] for graph-based semantic similarity [70] and word embeddings from word2vec [62] for corpus-based semantic similarity. The category2vec model [19] is proposed to learn vector representations of words and their categories jointly in a shared vector space without depending on labeled datasets. The study reports that graph-based semantic similarity approaches are better than corpus-based approaches when the contextual information is limited. Hence, semantic similarity features can be used to complement existing approaches.

Most previous studies have focused on unstructured text to learn representations of entities and neglect the useful structured data provided by the KB itself. However, [68] proposes a method that leverages the structured data in the KB for NED, using graph embeddings to integrate structured data from the KB with unstructured texts. The results suggest that graph embeddings learned from a graph of hyperlinks between Wikipedia articles can improve NN-based NED systems.

### 3) JOINT NER AND NED

Many previous studies have dealt with NER and NED in two separate steps. The consequence is that NED may not use all the information provided by NER [71], [72]. Hence, information about entity types and confidence [72] that might be useful for both tasks is not shared. Also, weak NER precision may decrease subsequent NED accuracy. Named Entity Recognition and Disambiguation (NERD) therefore deals with NER and NED jointly [24], [71], [72].

The JERL model [72] utilizes the mutual dependency between NER and NED. If NER’s confidence is high for both entity types and boundaries (contexts), it will encourage NED to link the consistent entity with NER’s outputs, and vice versa. The results show improvement of both tasks.

TwitterNEED [24] supports NERD of short and informal tweets. The extraction method focuses on high recall. An SVM is then used to filter out false positives using features from the disambiguation phase and the KB. For each extracted mention, a list of entity candidates is obtained from the YAGO KB along with the top-ranked pages from a Google query. An SVM is then used to rank the candidates based on context similarity features and a set of URLs. This increases accuracy without substantial harm to recall. Results are better compared to DBpedia Spotlight [13], Stanford NER, and the AIDA<sup>3</sup> disambiguation system. [73] presents an empirical study of NER and NED on short texts like tweets. J-NERD [71] uses a probabilistic graph model to perform joint NER and NED. It captures mention types, spans, and the linking of mentions to their entities in a KB. It then captures dependency parse trees for each sentence and derives both non-standard features about domains and novel features from the parse trees. In contrast to [24], information about uncertainty is retained for later steps.

Earlier joint methods, such as NERD-ML [74] (more below), relied on a dedicated extractor for each specific language. More recently, [75] has proposed a multilingual ensemble that combines multiple extractors for joint NER and NED, where the ensemble idea is to combine the output of several alternative components into a single and presumably better result, for example by averaging or voting. The ensemble produces a list of entities with their types and disambiguation links (called ground truths). The outputs of the NERD extractors are represented as real-valued vectors, which are then used as input to two ensemble neural networks (one for NER and one for NED).

### D. NAMED ENTITY LINKING

NEL annotates each mention in a text with the IRI of its corresponding entity as described in a KB in the LOD cloud. Although some NER approaches too annotate entity mentions, they are restricted to the type level, using a predefined set of a handful or a few dozen types such as persons, locations, organizations, and their subtypes. The number of entities available in a KG, on the other hand, can be in the millions. For example, DBpedia describes more than 14 million entities, and Wikidata more than 59 million.

Figure 6 illustrates the main lifting tasks on three input texts. First, NER extracts all the mentions in the text. Then NED ranks the disambiguation candidates. Finally, NEL maps each mention to an entity in the LOD cloud.

NEL’s mention-entity mapping can be considered a ranking problem that reduces the number of candidates by assigning a weight to each possible entity. The NED and NEL tasks are closely related: the former finds which entity a mention like “Bush” refers to, and the latter provides the LOD IRI for that entity. Differences are that NED does not have to deal with unambiguous mentions that refer to a unique entity and that NEL also has to deal with *NIL* entities

<sup>3</sup><https://github.com/codepie/aida>

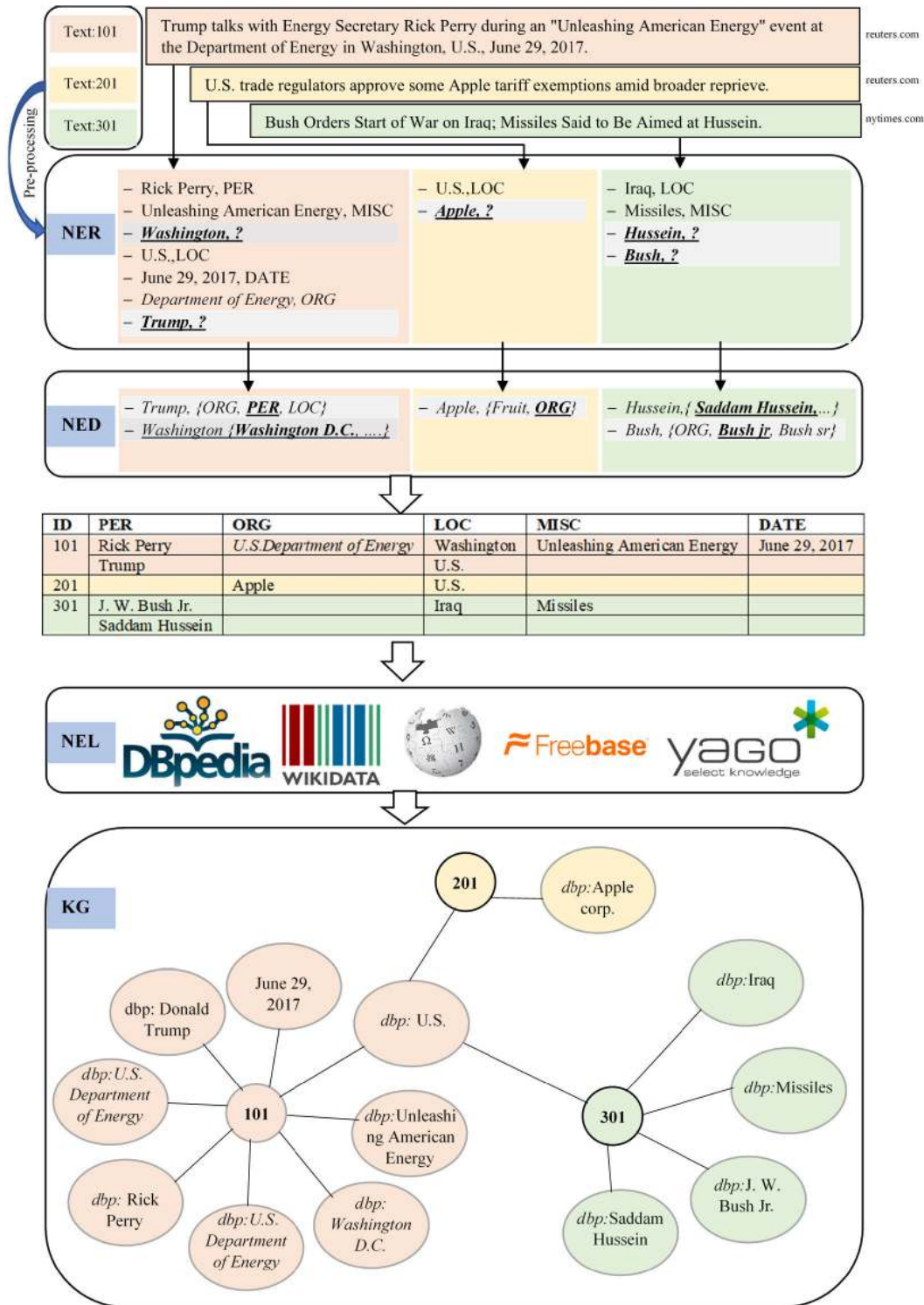


FIGURE 6. Named-entity extraction (NEE) of three example texts.

(also called *dark* or *emerging* entities) that have no entry in the KB. Nevertheless, many studies make no distinction between NED and NEL, and recent contributions have proposed joint-learning and end-to-end methods that perform NER, NED, and NEL together [19], [48].

### 1) TYPES, TASKS, AND TECHNIQUES

Early NEL approaches can be divided into three main types [76]: entity-by-entity methods [13], [77], [78], machine-learning methods [79], [80], and collective-linking methods [81]–[83]. The recent boom in neural networks

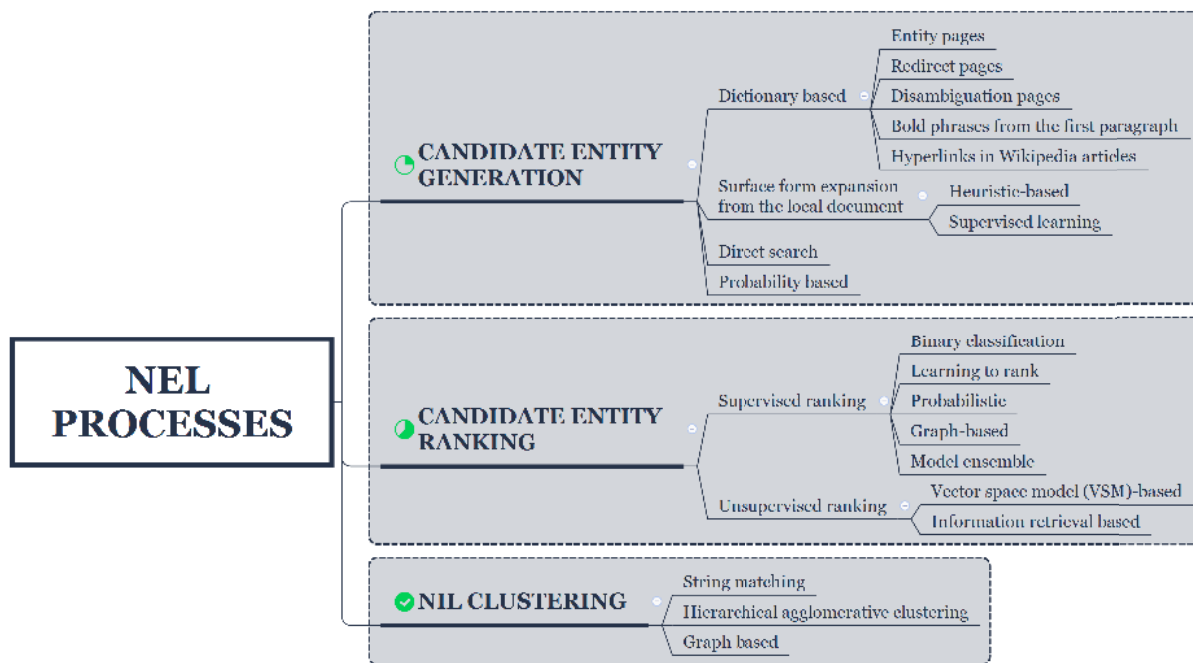


FIGURE 7. The three NEL sub-tasks along with their main techniques.

makes it more suitable to re-divide NEL approaches into three types: independent or feature-engineering methods, collective methods, and NN-based methods.

Regardless of type, NEL has three main sub-tasks: candidate-entity generation, candidate-entity ranking or disambiguation, and NIL clustering, where the middle step, candidate-entity ranking, resembles disambiguation. First, candidate-entity generation aims to retrieve all possible entities in the KB that may refer to an entity mention. Then, candidate-entity ranking aims to rank the candidate entities and return the most likely one for each targeted mention. Finally, NIL clustering deals with those mentions that cannot be matched with an entity in the KB [48]. We can group the majority of proposed approaches according to the NEL sub-tasks they cover, as shown in Figure 7. The figure also summarizes the main techniques proposed to tackle each step. As can be seen, NIL clustering has so far received less attention than the other sub tasks.

Although Figure 7 shows that many NEL approaches [84] have been proposed, few of them have dealt with all three sub-tasks. NEL approaches can therefore also be classified into: disambiguation-only methods and end-to-end methods. The disambiguation-only approaches focus on the second NEL step. They assume that all mentions and candidates have already been generated and link mentions to their corresponding entities in the KB. End-to-end and joint approaches deal directly with an input text and aim to extract all candidate mentions and link them to their corresponding entities in the KB. Most early contributions were disambiguation-only approaches, with end-to-end approaches being proposed more recently along with the

emergence of deep neural networks. Based on the literature we have reviewed [84], Figure 8 proposes a general model of NEL with an overview of the central sub-tasks and techniques.

### 2) PREVIOUS NEL RESULTS

Previous NEL results have been reported in [48], and [84] summarizes previous NEL research. [76] evaluated the most common NEL methods that were proposed between 2009 and 2014 and grouped them into three types (entity-by-entity, machine-learning, and collective-linking methods), excluding recent NN-based approaches. Entity-by-entity methods were proposed first, such as LIMES [77], Swoosh [78], Spotlight [13], AIDA-Light [85], and BEL [86]. Machine-learning methods like SHINE [79], GEML [87], and DNNEL [80] were proposed around the same time or a bit later. The most recent methods as of 2014 were collective-linking methods like AIDA [88], EMLC [89], LISTSVN [90], LGD2W [91], FHCEL [92], AGDISTIS [81], ELPH [82], and GSEL [83]. We reproduce Liu et al’s results [76] in Figure 9, which shows that the machine-learning and collective-linking based methods clearly outperform the entity-by-entity methods.

### 3) DISAMBIGUATION-ONLY NEL METHODS

Many empirical evaluations have been performed on state-of-the-art (SOTA) NEL methods using a variety of datasets. Figure 10 and Table 1 summarize the evaluation results for the most important methods on the most commonly used dataset: AIDA-CoNLL. [93], followed by [94]–[96], achieved the best



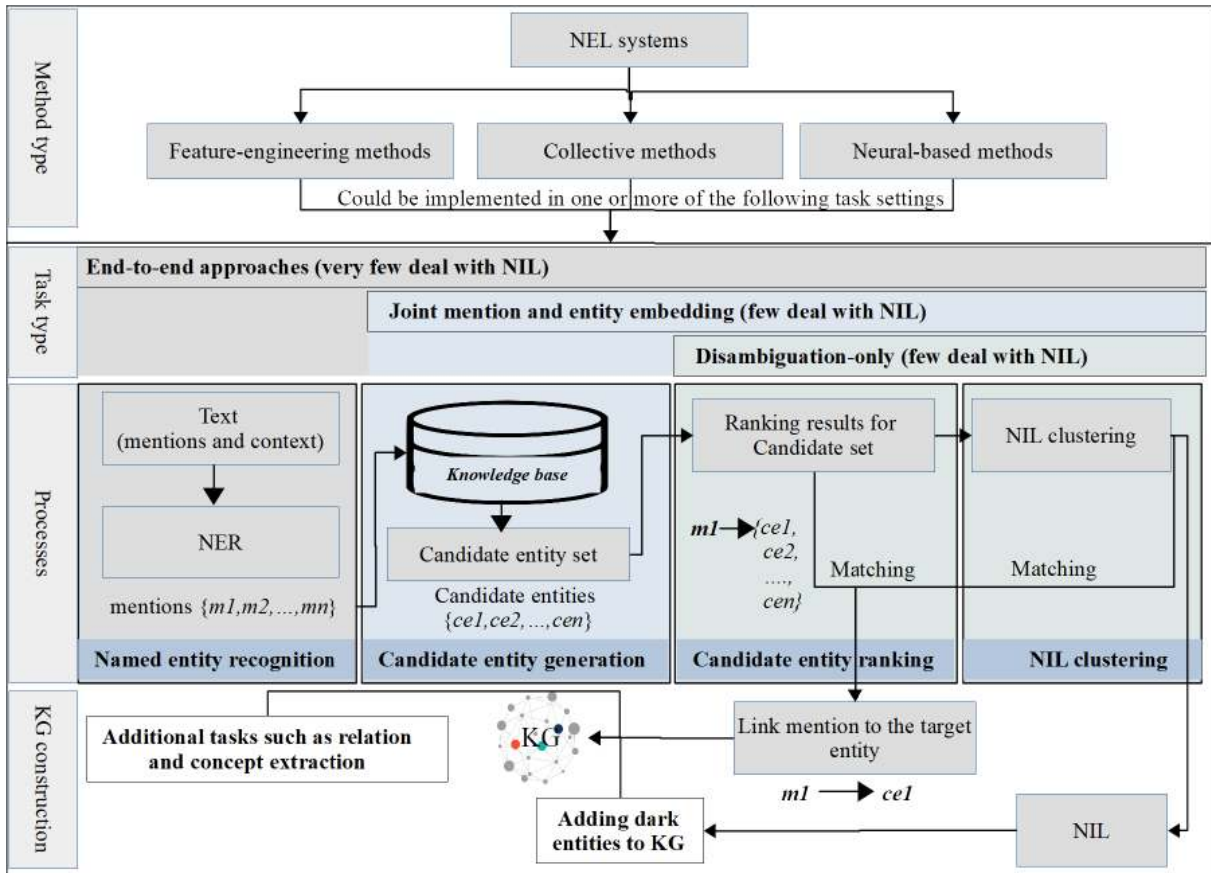


FIGURE 8. General NEL model along with central sub-tasks and techniques.

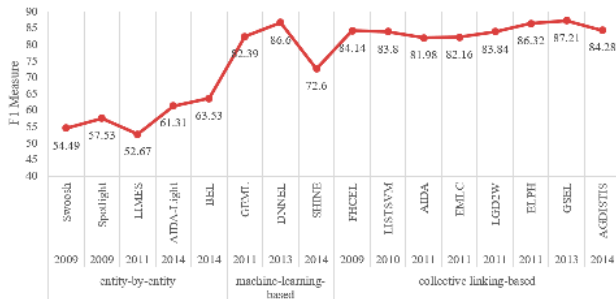


FIGURE 9. Early NEL methods as evaluated by [76].

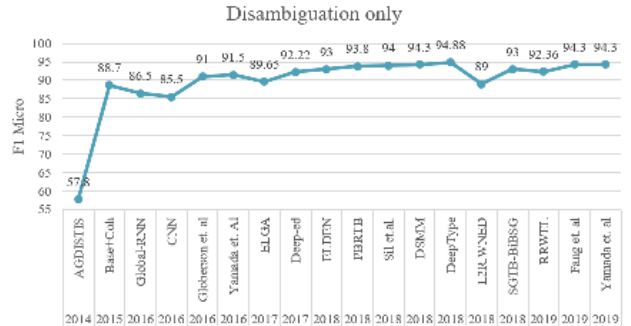


FIGURE 10. SOTA disambiguation-only results for NEL (AIDA-CoNLL).

results for disambiguation only; whereas [97] achieved the best end-to-end results followed by [98].

Table 2 shows that there is no perfect NEL model for all datasets. The best model for one dataset may perform poorly on others. An example is the SGTB-BiBSG model [99], which performed well on the WNED-CWEB dataset but not on the others. Only a small number of models performed best on more than one dataset. One example is WSRM [100], which was the best performer on two datasets (Reuters128 and RSS-500). Another is [94], which outperformed the others on the AQUAINT and ACE2004 datasets,

but not on any others. Models that perform well across many different types of dataset are called for.

The majority of studies neglect the relations between entity types and entity context. Joint mention and entity embeddings have therefore been proposed to take them into account [19]. An overview of early approaches to joint NER and NEL of long and short texts is provided in [101], [102]. Recognizing that NER and NEL were the focus of different research communities (i.e., of the NLP and the semantic-web communities, respectively), [101] proposes the NERD-ML approach to combine them. NERD-ML combines the strength

**TABLE 1. Recent NEL systems and results for the most commonly used dataset (AIDA-CoNLL).**

Year	Method	Publication Title	F1 Micro
2014	AGDISTIS	AGDISTIS — Graphbased Disambiguation of Named Entities Using Linked Data	57.8
2015	Base+Coh	Entity Disambiguation with Web Links	88.7
2016	Global-RNN	Joint Learning of Local and Global Features for Entity Linking via Neural Networks	86.5
2016	CNN	Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks	85.5
2016	Globerson et al.	Collective Entity Resolution with Multi-Focal Attention	91
2016	Yamada et al.	Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation	91.5
2017	ELGA	DBpedia-Based Entity Linking via Greedy Search and Adjusted Monte Carlo Random Walk	89.65
2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	92.22
2018	ELDEN	ELDEN: Improved Entity Linking Using Densefied Knowledge Graphs	93
2018	PBRTB	Bilinear Joint Learning of Word and Entity Embeddings for Entity Linking	93.8
2018	Sil et al.	Neural Cross-Lingual Entity Linking	94
2018	DSMM	Deep Semantic Match Model for Entity Linking Using Knowledge Graph and Text	94.3
2018	DeepType	DeepType: Multilingual Entity Linking by Neural Type System Evolution	<b>94.88</b>
2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	89
2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	93
2019	RRWEL	Neural Collective Entity Linking Based on Recurrent Random Walk Network Learning	92.36
2019	Fang et al.	Joint Entity Linking with Deep Reinforcement Learning	94.3
2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	94.3

**TABLE 2. Recent disambiguation-only results for the most commonly used NEL datasets.**

Dataset	Year	Model name	Publication Title	F1 Micro
ACE2006	2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	88
	2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	89.2
	2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	<b>91.5</b>
AQUAINT	2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	87
	2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	90.5
	2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	<b>93.7</b>
MSNBC	2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	92
	2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	92.6
	2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	<b>93.8</b>
Reuters128	2011	AIDA	Robust Disambiguation of Named Entities in Text	56.6
	2014	AGDISTIS	AGDISTIS — Graphbased Disambiguation of Named Entities using linked data	68.9
	2014	Babelify	Entity Linking Meets Word Sense Disambiguation: a Unified Approach	54.8
	2019	WSRM	Using Knowledge Base Semantics in Context-Aware Entity Linking	<b>79.9</b>
RSS-500	2011	AIDA	Robust Disambiguation of Named Entities in Text	65.5
	2014	AGDISTIS	AGDISTIS — Graphbased Disambiguation of Named Entities Using Linked Data	54.2
	2014	Babelify	Entity Linking Meets Word Sense Disambiguation: a Unified Approach	64.1
	2019	WSRM	Using Knowledge Base Semantics in Context-Aware Entity Linking	<b>79.3</b>
WNED-CWEB	2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	77
	2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	<b>81.8</b>
	2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	74.9
WNED-WIKI	2018	L2R.WNED	Robust Named Entity Disambiguation with Random Walks	<b>84.5</b>
	2018	SGTB-BiBSG	Collective Entity Disambiguation with Structured Gradient Tree Boosting	79.2
	2019	Yamada et al.	Pre-training of Deep Contextualized Embeddings of Words and Entities for Named Entity Disambiguation	83.9

of crowd-entity extractors with web-entity recognizers and machine learning.

Some previous contributions used frame semantics along with semantic role labelling, such as PIKES [103] and FRED [104]. The PIKES system [103] for extracting knowledge from NL text has two phases. First, several NLP techniques are combined to integrate entity mentions into a single RDF graph. Then, the resulting mentions graph is processed with SPARQL-like mapping rules to produce a KG organized around semantic frames.

FRED [104] is a frame-based machine reader that produces OWL/RDF representations of texts. It integrates the

outputs of existing NLP tools and LOD resources such as Boxer [105], BabelNet<sup>4</sup>, and DBpedia to expand extracted tacit knowledge. FRED uses TagME<sup>5</sup>, which uses deep sentence parsing for NER and NEL, combined with Wikipedia context for NED. Although FRED [106] is reported to be flexible and usable without specific tuning, the reported precision is not high. Reference [107] proposes a hybrid approach for NER and NEL that outperforms FRED [106] by 20% in F-measure. The approach combines linguistic and semantic

<sup>4</sup><http://babelnet.org/>

<sup>5</sup><http://tagme.di.unipi.it/>

**TABLE 3.** Recent NEL end-to-end systems and results for the most commonly used dataset (AIDA-CoNLL).

Year	Method	Publication Title	F1 Micro
2014	WAT	From TagME to WAT: a New Entity Annotator	69.32
2014	Babelfy	Entity Linking meets Word Sense Disambiguation: a Unified Approach	71.8
2015	PBoH	Probabilistic Bag-Of-Hyperlinks Model for Entity Linking	87.39
2016	J-nerd	J-nerd: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features	78.7
2018	NCEL	Neural Collective Entity Linking	80
2018	Kolitsas et al.	End-to-End Neural Entity Linking	<b>89.4</b>
2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	<b>93.07</b>
2019	Martins et al.	Joint Learning of Named Entity Recognition and Entity Linking	81.9

features and uses them to recognize and index IRIs from DBpedia. The aim is to increase recall in the recognition task and then prune candidates later.

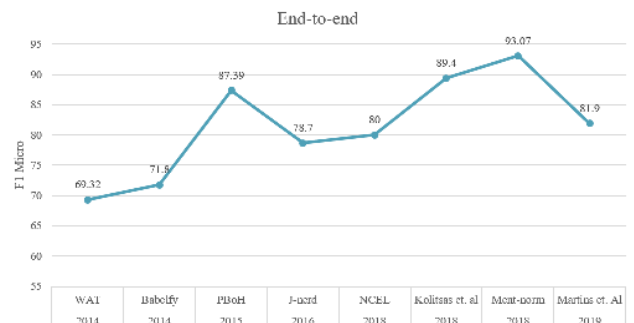
Entity Extraction and Linking (EEL) [20] is an OpenIE-based approach that employs thematic roles to link relation phrases with known properties used on the semantic web by integrating an ensemble of alternative NER and NEL systems. EEL handles both cases where entities are duplicated because they have the same text fragment and IRI and cases where two or more entities are overlapping because they share the same text fragment [20]. The authors demonstrate that using multiple entity-linking systems improve extraction accuracy. Moreover, the named entities associated with noun phrases maintain the coherency of RDF data.

#### 4) END-TO-END NEL METHODS

The majority of previous studies assumed that mentions and entities were already available and focused only on the disambiguation process, neglecting mutual dependencies between mentions and their entities. But a practical solution must cover all extraction phases. To overcome this, end-to-end entity linking has received increasing attention recently, inputting raw text and aiming to extract all its mentions and link them to their entities in a KB. Although few end-to-end studies have been published so far [53], [66], [71], [108], interesting recent examples include NN-based end-to-end linking models such as [97], [98], [109], [110].

Reference [98] proposed the first NN-based end-to-end linking system to do joint mention detection and entity disambiguation, in order to capture the dependency between both tasks and reduce propagation of errors from NER to NEL. They consider all word spans that might mention an entity and use word and entity embeddings to compute a context-aware compatibility score for ranking the candidates. The authors demonstrate that engineered features are almost unnecessary when using end-to-end approaches. Their model reaches SOTA results on the AIDA/CoNLL dataset and, when combined with Stanford NER, it generalizes well to other datasets with different characteristics.

The end-to-end, trainable Neural Collective Entity Linking (NCEL) model [109] addresses the data sparsity of local contextual features when resolving entities independently. It applies Graph Convolutional Networks on sub-graphs of the entity graph. NCEL thus learns features from both local and global information. It is trained on Wikipedia

**FIGURE 11.** SOTA disambiguation-only results for NEL (AIDA-CoNLL).

hyperlinks using an attention mechanism to deal with noisy data.

Ment-norm [97] is an end-to-end system for NEL that considers relations as latent. Representation learning was used to learn relation embeddings, eliminating the need for extensive feature engineering.

[110] proposes a Stack-LSTM network model for joint NER and NEL. Sharing information between the NER and NEL tasks, NER suggests the entity type (e.g., ORG, PER, LOC) which is then used to better disambiguate the entity during NEL. The results of using the joint model are compared to using NER and NEL separately, showing that the joint model outperforms the individual ones.

Tables 3 and 11 summarize the most relevant end-to-end NEL approaches and results for the most commonly used dataset (AIDA-CoNLL). The best end-to-end results were produced by the approach presented in [97], followed by [98]. Many other evaluations have been done using other datasets, as shown in Table 4. As for the disambiguation-only approaches, there is no perfect method for all datasets, and approaches that produce good results on a particular dataset do not perform well on others. Examples are the NCEL model [109] that outperforms others on the WNED-WIKI dataset, and Deep-ed [66] that reports good results for AQUAINT. Moreover, Ment-norm [97] is the best end-to-end model for the MSNBC and ACE2004 datasets, as shown in Table 4. Possible reasons include the nature of the datasets and the variation of the type and amount of data used to train each model. Developing models that perform well across different types of datasets remains a challenging task.

Recent NN-based approaches for NEL using joint embedding and end-to-end learning have performed best so far.

**TABLE 4.** Recent end-to-end results for the most commonly used NEL datasets.

Dataset	Year	Model name	Publication Title	F1 Micro
ACE2004	2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	88.5
	2018	NCEL	Neural Collective Entity Linking	88
	2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	<b>89.9</b>
AQUAINT	2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	<b>88.5</b>
	2018	NCEL	Neural Collective Entity Linking	87
	2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	88.3
MSNBC	2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	93.7
	2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	<b>93.9</b>
Reuters128	2016	PBoH	Probabilistic Bag-of-hyperlinks Model for Entity Linking	<b>68.3</b>
	2018	Kolitsas et al.	End-to-end neural entity linking	54.6
RSS-500	2016	PBoH	Probabilistic Bag-of-hyperlinks Model for Entity Linking	<b>55.3</b>
	2018	Kolitsas et al.	End-to-end Neural Entity Linking	42.2
WNED-CWEB	2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	<b>77.9</b>
	2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	77.5
WNED-WIKI	2017	Deep-ed	Deep Joint Entity Disambiguation with Local Neural Attention	77.5
	2018	NCEL	Neural Collective Entity Linking	<b>86</b>
	2018	Ment-norm	Improving Entity Linking by Modeling Latent Relations between Mentions	78

**TABLE 5.** The most used datasets for NEL.

Dataset	Type	Doc.length	Format	Encoding	License	Mentions	KB
AIDA-CoNLL	news	medium	TSV	ASCII	Open	34587	YAGO/Freebase/Wikipedia
NEEL	tweets	short	TSV	UTF8	Open	8665	Dbpedia
OKE2015	encyclopedia	long	NIF/RDF	UTF8	Open		Dbpedia
RSS-500	news	medium	NIF/RDF	UTF8	Open	1000	Dbpedia
WES2015	blog	long	NIF/RDF	UTF8	Open		Dbpedia
WikiNews	news	medium	stand-off XML	UTF8	Open		Dbpedia
KORE50	tweets	short	NIF/RDF	UTF8	Open	143	YAGO/Dbpedia
OKE2016	Web pages	long	NIF/RDF	UTF8	Open	1043	Dbpedia
AQUAINT	news	medium	NIF/RDF	UTF8	Closed	449	Wikipedia
TAC-KBP2011	news	medium	NIF/RDF	UTF8	Source data via registration	4338	Wikipedia
TAC-KBP 2016	news	medium	NIF/RDF	UTF8	Source data via registration	9231	Wikipedia
TAC-KBP 2017	news	medium	NIF/RDF	UTF8	Source data via registration	6915	Wikipedia
Reuters128	economic news	medium	NIF/RDF	UTF8	Open	881	Wikipedia
ACE2004	news	medium	XML	UTF8	Closed	257	Wikipedia
MSNBC	news	medium	NIF/RDF	UTF8	Open	656	Wikipedia
WNED-CWEB (CWEB)	news	medium	NIF/RDF	UTF8	Open	11,154	Wikipedia
WNED-WIKI (WW)	news	medium	NIF/RDF	UTF8	Open	6,821	Wikipedia

For real-word applications with more open domains and dynamic environments, end-to-end approaches look promising, although there are issues that need to be considered. One issue is that their accuracy drops dramatically when trained on or applied to small datasets [98], so that extensive datasets are required for robust models. Other issues are the training of contextual embeddings for entities and extending the approaches to become cross-lingual [110]. Furthermore, the end-to-end approaches have not yet taken the NIL problem sufficiently into account.

##### 5) DATASETS AND KBS USED FOR NEL

Many NEL datasets have been published and used in previous studies. Table 5 shows the NEL datasets we have encountered [84], [94], [100], [111].

Several popular LOD KBs have been used as a targets of NEL systems. The most popular ones are Dbpedia, Freebase, Wikidata, and Wikipedia. Dbpedia is a semantic (RDF)

extract of Wikipedia and the two contains almost the same entities [76], [84], [111]. Wikidata is the Wikimedia foundation's knowledge graph, intended as a central, crowd-sourced knowledge base for feeding Wikipedia's fact boxes and other Wikimedia projects. Freebase is an older and now defunct large, crowd-sourced knowledge graph.

#### IV. DISCUSSION

Many approaches have been proposed for all or some of the tasks involved in lifting NL texts to KGs. In the early stages, NLP has played a prominent role, providing techniques for dealing with NL text, including various pre-processing tasks. NLP techniques have also been used for NER, which recognizes and classifies mentions in the text. In the later stages, AI, ML, KG and LOD techniques have become important. They are used by NED- and NEL-methods to disambiguate the mentions and link them to knowledge-base resources that represent entities. Recent deep-NN techniques have covered both the early and late stages.

## A. LANGUAGES

The majority of the papers have lifted English language, although we have found a small number of studies that consider languages such as Italian [25], [112], French [22], Chinese [113], Spanish [112], [114], Malay [115], Romanian [116], and Russian [117].

We have only found a few studies that investigate more than one language: NewsReader [112] which considers English, Dutch, Spanish and Italian; [118] which considers English and Dutch; and [119] which considers morphological languages such as Turkish and Russian. Hence, dealing with multilingual texts in general has so far received little attention [75], perhaps in part due to the limited availability of training datasets for languages other than English. Recently proposed multi- and cross-lingual approaches such as [112], [119] can train a model on a richly-resourced language and then apply it to a more sparsely-resourced one. Other approaches to multi- and cross-lingual NEL [93], [120]–[122] need to be re-trained for every new language and, compared to the single-language approaches, accuracy is still weak [121]–[123]. One reason may be that environment and structure change from one language to another. Another possible reason is that some of these studies, such as [123], use automatic translation from the source to the target language in order to benefit from its available resources, making their accuracy dependent on translation quality. Most of the proposed studies focus on either NER, NED, or NEL [121]. A cross-lingual approach that covers all parts of named-entity extraction is called for [124].

## B. DOMAINS

The majority of previous studies have focused on general news due to the abundance of datasets [23], [112], [125], and a few of them focus on news for specific domains such as IT [20] and sports [126]. Besides news, there is research dedicated to domains such as software and source code [127]–[129], cyber security [130], and reviews [23]. Despite the abundance of training data available (albeit mainly for English), the datasets are restricted to certain domains. Hence, establishing a stable NL lifting system for a new domain is effortful and expensive. Only a few papers [124], [131]–[133] have investigated transfer learning and domain adaptation, in which the model makes use of labelled training datasets available in a richly-resourced domain (such as news) and transfers the learning to recognize entities in another, more poorly-resourced domain (such as biomedicine). The few systems proposed so far suffer from low accuracy.

## C. TEXT TYPES

In general, most of the proposed methods are optimized for either short or long texts. Few studies have considered both. The limited set of papers that discussed short texts like tweets [19], [23], [24], [134], micro posts [125], [135],

SMS texts [22], and chats [27] have reported that dealing with shorter texts is harder and more challenging [22].

Only a very limited number of published studies have dealt with real-time data and even fewer have been run in a real-time environment [21].

## D. EVALUATION

Lack of standard evaluation methods makes it difficult to encourage improvement of the field based on systematic comparison the strengths and weaknesses of the different approaches. Although comparing approaches across tasks, languages and domains will always be difficult, repeatable and reliable methods are required to evaluate and compare methods that have similar aim and scope.

Concerning evaluation tasks, four aspects need to be considered: the ability of the system to recognize entity mentions; the ability of the system to correctly assign types to mentions; the ability of the system to identify the intended entity referred to by a mention; and the ability of the system to link mentions to entities in a KB.

Concerning experimental set up, only a few studies have used the GERBIL [125], [136] framework for benchmarking NER and NEL systems. GERBIL outputs comparable evaluation results that can help researchers and developers discover the strengths and weaknesses of their approaches compared to the state of the art. It tackles the main issue of NEL evaluation and clarifies how two IRIs could be compared to each other and evaluated without being limited to a particular KB. However, GERBIL does not offer any additional explanations (such as error analysis) for each mention [120].

Concerning gold standards, the majority of NER and NEL models are evaluated using manually-created gold standards, which has some drawbacks according to [137]. First, the current gold standards do not share a common set of rules concerning what should be recognized and linked as an entity. Second, most gold standards frequently include mistakes because they have not been verified by other scholars. Third, many gold standards become outdated when the reference KBs that entities have been linked to evolve over time but the gold standards are not updated to reflect the most recent version of the reference KB [137]. [84] points out that, to evaluate NEL on non-popular entities or on specific domains, benchmark datasets that link to domain-specific resources and long-tail entities are needed.

Concerning validation, although tenfold cross validation is most commonly used in NLP applications, some studies have used different cross-validation settings [24]. This makes comparison more difficult and potentially unfair.

Concerning metrics, most of the selected studies have used the standard evaluation metrics for NLP tasks: precision, recall, F-measure, and accuracy.

## E. NLP TOOLS AND APIS

Many NLP tools and APIs have been used in NL-to-KG pipelines.

TABLE 6. Available tools and systems (OS: open source).

Year	System name	NER	NED	NEL	URL
2002	NLTK — (OS)	Y	N	N	<a href="https://www.nltk.org/">https://www.nltk.org/</a>
2003	Cicerolite- (Private)	Y	N	N	<a href="http://www.languagecomputer.com/products/text-annotation/cicerolite.html">http://www.languagecomputer.com/products/text-annotation/cicerolite.html</a>
2005	AlchemyAPI (now available with IBM Watson NLU)	Y	N	N	<a href="https://cloud.ibm.com/apidocs">https://cloud.ibm.com/apidocs</a>
2010	Tagme — (OS)	Y	Y	Y	<a href="https://tagme.d4science.org/tagme/">https://tagme.d4science.org/tagme/</a>
2011	DBpedia Spotlight — (OS)	Y	Y	Y	<a href="https://www.dbpedia-spotlight.org/">https://www.dbpedia-spotlight.org/</a>
2011	Ambiverse (AIDA previously) — (OS)	Y	Y	Y	<a href="https://github.com/codepie/aida">https://github.com/codepie/aida</a>
2011	Fox — (OS)	Y	N	N	<a href="http://aksw.org/Projects/FOX.html">http://aksw.org/Projects/FOX.html</a>
2011	The OrganismTagger System — (OS)	Y	Y	Y	<a href="http://www.semanticsoftware.info/organism-tagger">http://www.semanticsoftware.info/organism-tagger</a>
2012	FRED — (OS)	Y	Y	Y	<a href="http://wit.istc.cnr.it/stlab-tools/fred/">http://wit.istc.cnr.it/stlab-tools/fred/</a>
2013	GATE — (OS)	Y	N	N	<a href="https://gate.ac.uk/">https://gate.ac.uk/</a>
2013	TwitIE -(OS)	Y	Y	Y	<a href="https://gate.ac.uk/wiki/twitie.html">https://gate.ac.uk/wiki/twitie.html</a>
2014	Stanford CoreNLP	Y	N	N	<a href="https://stanfordnlp.github.io/CoreNLP/">https://stanfordnlp.github.io/CoreNLP/</a>
2014	Europeana newspapers — (OS)	Y	N	N	<a href="http://www.europeana-newspapers.eu/named-entity-recognition-for-digitised-newspapers">http://www.europeana-newspapers.eu/named-entity-recognition-for-digitised-newspapers</a>
2014	TextBlob — (OS)	Y	N	N	<a href="https://textblob.readthedocs.io/en/dev/">https://textblob.readthedocs.io/en/dev/</a>
2015	PIKES — (OS)	Y	Y	Y	<a href="http://pikes.fbk.eu/">http://pikes.fbk.eu/</a>
2015	YODIE — (OS)	Y	Y	Y	<a href="https://gate.ac.uk/applications/yodie.html">https://gate.ac.uk/applications/yodie.html</a>
2015	GATE Mimir — (OS)	Y	Y	Y	<a href="https://gate.ac.uk/mimir/">https://gate.ac.uk/mimir/</a>
2015	Polyglot — (OS)	Y	N	N	<a href="https://sites.google.com/site/rmyeid/projects/polyglot-ner">https://sites.google.com/site/rmyeid/projects/polyglot-ner</a>
2015	SpaCy — (OS)	Y	Beta	Beta	<a href="https://spacy.io/">https://spacy.io/</a>
2016	PKE — (OS)	Y	N	N	<a href="http://www.aclweb.org/anthology/C16-2015">http://www.aclweb.org/anthology/C16-2015</a>
2018	NLP-Cube — (OS)	Y	N	N	<a href="http://www.aclweb.org/anthology/K18-2017">http://www.aclweb.org/anthology/K18-2017</a>
2018	GATE Bio-YODIE — (OS)	Y	Y	Y	<a href="https://gate.ac.uk/applications/bio-yodie.html">https://gate.ac.uk/applications/bio-yodie.html</a>
	Apache OpenNLP — (OS)	Y	N	N	<a href="http://opennlp.apache.org/">http://opennlp.apache.org/</a>
	Apache Stanbol — (OS)	Y	N	N	<a href="https://stanbol.apache.org/">https://stanbol.apache.org/</a>

For pre-processing, the Stanford Core NLP [12] is the most commonly used [20]. Other tools include spaCY<sup>6</sup>, TextBlob<sup>7</sup>, NLTK<sup>8</sup>, OpenNLP [138], the Pangloss NLP pipeline [27], TEXTPRO for Italian language [25], GATE ANNIE [21], and NLP-Ce [28]. Most of them are restricted to a single or a limited number of languages.

Many other tools for NER, NED, and NEL have been proposed, such as AlchemyAPI (now available with IBM Watson NLU)<sup>9</sup>, DBpedia Spotlight [13], Stanford CoreNLP [12], PIKES [103], YODIE [139], GATE [140], GATE Mimir [7], CiceroLite, FOX [141], FRED [104], PKE [142], and OpenCalais<sup>10</sup>. A comparison of established tools up to 2013 is provided in [143]. Systems like Babelify [53], AIDA [88], and DBpedia Spotlight [13] tend to prioritize recall and do little to prune likely false positive candidates [107]. Hence, a strong NER system is needed to ensure good precision [107].

[102] offers an empirical evaluation of the robustness of previous systems such as GATE ANNIE [10], Stanford NER [12], YODIE [144], NERD-ML [74], Alchemy API, DBpedia Spotlight [13], Lupedia<sup>11</sup>, TextRazor<sup>12</sup> and Zemanta<sup>13</sup>. The results show that hybrid approaches tend to perform better. Recall in previous systems evaluations may have been underestimated because most of them link

to DBpedia, which does not contain entities that match all mentions.

In view of the research published on NER, NED, and NED, the available tools have not kept pace with the most recent developments. They also mostly focus on a single task, such as NER or NED, with only a few covering more than one. Moreover, most of the tools are built for only one or a limited number of languages, and they are hard to adapt to new settings, among other things due to: dependence on commercial tools; choice of programming language; code that is hard to understand and extend; and tight coupling with a particular KB. Hence, it can be challenging to adapt existing systems to build domain-specific applications [21]. According to [120], each tool constructs its KG differently. Consequently, it is challenging to utilize those tools for linking from different KBs. Table 6 lists the most commonly used tools along with the tasks they support.

Recently, [145] have evaluated three of the most commonly used tools (Ambiverse (known as AIDA previously), Babelfy, and TagMe) and reported that Babelfy and Ambiverse achieved the best result with slightly lower recall in Ambiverse. Nevertheless, their precision were at most 81% while their recall did not exceed 68%. Although the study did not include a popular system like DBpedia Spotlight, the results suggest that these systems still perform far from the state of the art.

Figure 12 shows the results of evaluating 15 common tools over a wide selection of available datasets using GERBIL<sup>14</sup>. Figure 12 shows the results with strong annotation match setting, while Figure 13 depicts the results with weak annotation match.

<sup>6</sup><https://spacy.io/><sup>7</sup><https://textblob.readthedocs.io/en/dev/><sup>8</sup><https://www.nltk.org/><sup>9</sup><https://cloud.ibm.com/apidocs><sup>10</sup>[www.opencalais.com](http://www.opencalais.com)<sup>11</sup><http://lupedia.ontotext.com><sup>12</sup><http://www.textrazor.com><sup>13</sup><http://www.zemanta.com>

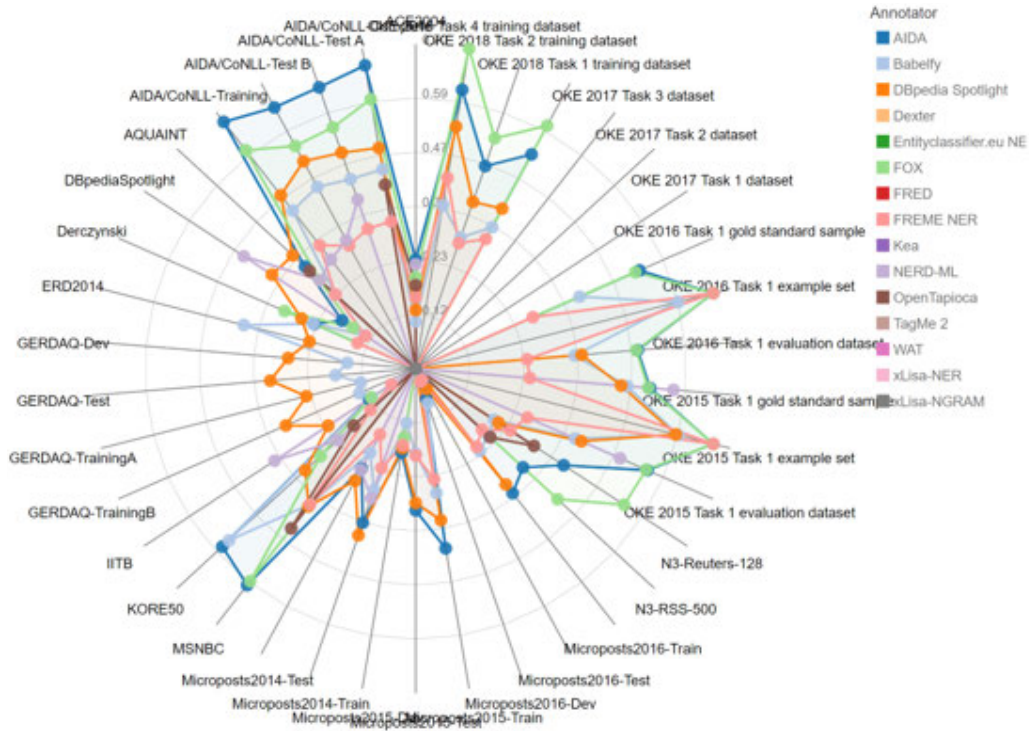


FIGURE 12. Evaluation of tools using GERBIL – strong annotation.

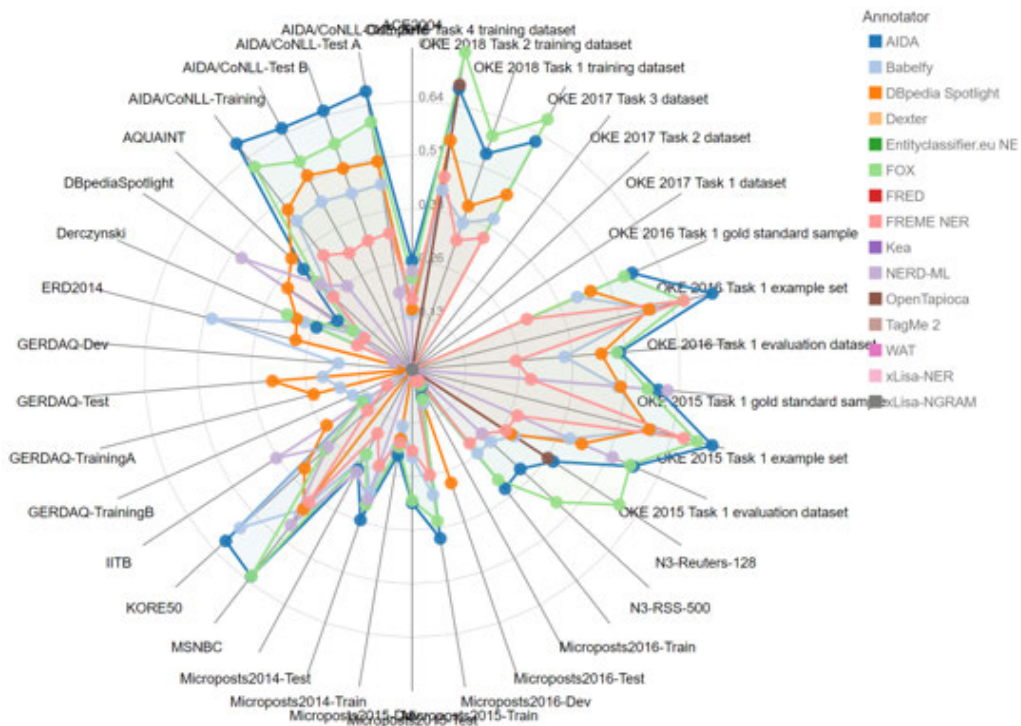


FIGURE 13. Evaluation of tools using GERBIL – weak annotation.

**F. MAIN FINDINGS AND OPEN DIRECTIONS**

NER results impact later tasks such as NED and NEL [146], [147], which in turn are highly affected by the type and quality of the documents used for

training [146]. KG researchers have been focusing more on the KG-construction side and have paid less attention to the NER task, which is usually done by off-the-shelf systems or established tools. They integrate such NER tools mostly

internally using the available APIs in their code. With such integration, re-configuring those external tools or switching between them becomes difficult [47]. Despite recent deep NN-based approaches improving the state-of-the-art NER results, they have not yet been leveraged to lift NL to KG as mentioned earlier.

Most NER systems are designed for a limited set of entity types (usually people, organizations and locations) and cannot be easily reconfigured to support other types of entities [20], [34]. Transfer-learning approaches that tackle this shortcoming is a promising direction for future research [148], [149].

Approaches to NED can be classified into individual, collective, and recent NN-based approaches. Of these, the NN-based approaches appear to be most effective [51], [58].

Our paper has defined NEL as a wider task that includes NED as one of its sub-tasks, but this is not a generally accepted distinction, and many researchers are using the two initialisms synonymously. Many NEL approaches use off-the-shelf systems for NER. However, choosing the best NER-model to use for NEL is still a challenging task due to the hardness of comparing the dataset used to train the system and the dataset that needs to be processed [47]. Unless end-to-end methods are used, obtaining good NEL results requires improving and making use of the most recent NER approaches [120].

Reference [120] discusses several NEL issues. One is that gold standards can contain wrong annotations (such as U.K instead of UK) and missing annotation links. Another concern is that KBs can contain bad redirects. Moreover, most NEL tools confuse regions and cities that have the same names (e.g., New York State or New York City, and Valencia the region, province, or city).

Recent studies indicate that the accurate tuning of several steps, especially tokenization and semantic similarity, is necessary for robust NED and NEL systems [27]. Deep NN-based NLP methods [40]–[44] currently outperform the earlier approaches [28], [147].

[150] argue that all recent NEL models neglect three essential points. The first one is that mentions and entities are correlated, particularly in semantic features. The second one is that entities and relations are equally important and tightly connected basic units in the KB. The third one is that the attention mechanisms in recent neural-network approaches, which can potentially filter out irrelevant information from contexts, have been neglected in entity-embedding construction [150].

NEL difficulty varies not only between datasets but also between mentions, where datasets and mentions that are difficult to link often share characteristics like short documents, many entities per document, many salient entity types, and many entities in total [145].

Recent contributions that lift NL into KGs demonstrate that the integration of entity-extraction and -linking systems improves extraction accuracy [20]. Moreover, associating named entities with more descriptive noun phrases

preserves meaning when combining entities from multiple sources. [20]. However, although the disambiguation-only NEL methods tend to beat end-to-end approaches today, the latter may be a more promising direction with potential for further improvement. End-to-end NEL is particularly promising for open domains where there are no gold standards available.

Reference [151] calls for a genuine semantic-web reference evaluation framework to assist the research community. Lifting approaches are usually assessed using tasks that do not focus on specific semantic web and KG aims. Of course, general tasks such as NER, NED, NEL, and relation extraction are important, but they are usually designed without evaluating the output as knowledge graphs, Linked Data or OWL ontologies [151].

## V. CONCLUSION

The paper has presented an overview of recent advances in a central task for lifting NL texts to KGs: that of named-entity extraction. We see this as an important step towards making the abundance of NL information on the internet available as computer-processable KGs. We conclude with a few general observations: Many approaches for lifting NL to KG are based on previous-generation NER methods, and new lifting approaches are needed that add disambiguation and linking to best-of-breed NER techniques. There is also a lack of standards for comparing extraction approaches. This can partly be attributed to a lack of commonly accepted evaluation methods, but it also a consequence of the recognition-disambiguation-linking pipeline. For example, it is hard to fairly compare pure NER with combined NER-NED-NEL approaches when the latter is restricted to identifying named entities in the KB that is used for disambiguation and linking.

NEL has moved from being a pipeline of clear and isolated steps into an integrated process in two main ways. The first is that traditional sequential steps are now being integrated by joint learning and end-to-end processes. The second is that mentions and entities that were previously analysed in isolation are now being lifted in each other's context. The current culmination of these trends are the deep-learning approaches that have reported promising results in recent years.

In our future work we plan to develop complementary overviews that also cover lifting of general concepts and of relations between entities. Many recent approaches also lift relations jointly with entities (both named entities and concepts), emphasising the need for a unified lifting framework that is not restricted to named entities. A comparative evaluation of state-of-the-art methods in the same environment using the same settings and datasets is another research direction.

## REFERENCES

- [1] A. Singhal. (2012). *Introducing the Knowledge Graph: Things, not Strings*. [Online]. Available: <https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>



- [2] L. Ehrlinger and W. Wöb, "Towards a definition of knowledge graphs," in *Proc. Joint Posters Demos Track 12th Int. Conf. Semantic Syst. (SEMANTICS)*, 1st Int. Workshop Semantic Change Evolving Semantics (SuCESS) Co-located 12th (SEMANTICS), 2016, p. 4. [Online]. Available: <http://ceur-ws.org/Vol-1695/paper4.pdf>
- [3] J. Yan, C. Wang, W. Cheng, M. Gao, and A. Zhou, "A retrospective of knowledge graphs," *Frontiers Comput. Sci.*, vol. 12, no. 1, pp. 55–74, Feb. 2018.
- [4] R. Guha, D. Brickley, and S. Macbeth, "Schema.Org: Evolution of structured data on the Web," *Commun. ACM*, vol. 59, no. 2, pp. 44–51, 2016.
- [5] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Amsterdam, The Netherlands: Elsevier, 2011.
- [6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 28–37, 2001.
- [7] V. Tablan, K. Bontcheva, I. Roberts, and H. Cunningham, "Mimir: An open-source semantic search framework for interactive information seeking and discovery," *J. Web Semantics*, vol. 30, pp. 52–68, Jan. 2015.
- [8] A. Perez, R. Basagoiti, R. A. Cortez, F. Larrinaga, E. Barrasa, and A. Urrutia, "A case study on the use of machine learning techniques for supporting technology watch," *Data Knowl. Eng.*, vol. 117, pp. 239–251, Sep. 2018.
- [9] G. G. Chowdhury, "Natural language processing," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, 2005.
- [10] H. Cunningham, "Gate: A framework and graphical development environment for robust NLP tools and applications," in *Proc. 40th ACL (ACL)*, 2002, pp. 168–175.
- [11] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proc. ACL Workshop Effective Tools Methodol. Teach. Natural Lang. Process. Comput. Linguistics*, 2002, pp. 63–70.
- [12] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd ACL Syst. Demonstrations*, 2014, pp. 55–60.
- [13] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proc. 9th Int. Conf. Semantic Syst.*, 2013, pp. 121–124.
- [14] K. Gashteovski, R. Gemulla, and L. del Corro, "MinIE: Minimizing facts in open information extraction," in *Proc. Conf. EMNLP*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2630–2640. [Online]. Available: <https://www.aclweb.org/anthology/D17-1278>
- [15] M. Dürst and M. Suignard, *Internationalized Resource Identifiers (IRIs)*, document RFC 3987, 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc3987>
- [16] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, "Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO," *Semantic Web*, vol. 9, no. 1, pp. 77–129, 2018. [Online]. Available: <http://www.semantic-web-journal.net/system/files/swj1465.pdf>
- [17] S. Staab and R. Studer, *Handbook on Ontologies* (International Handbooks on Information Systems). Berlin, Germany: Springer-Verlag, 2009.
- [18] C. Bizer, T. Heath, Jr., and T. Berners-Lee, "Linked data: The story so far," in *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. Harrisburg, PA, USA: IGI Global, 2011, pp. 205–227.
- [19] G. Zhu and C. A. Iglesias, "Exploiting semantic similarity for named entity disambiguation in knowledge graphs," *Expert Syst. Appl.*, vol. 101, pp. 8–24, Jul. 2018.
- [20] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, "OpenIE-based approach for Knowledge Graph construction from text," *Expert Syst. Appl.*, vol. 113, pp. 339–355, Dec. 2018.
- [21] P. Fafalios, M. Baritakis, and Y. Tzitzikas, "Exploiting linked data for open and configurable named entity extraction," *Int. J. Artif. Intell. Tools*, vol. 24, no. 02, Apr. 2015, Art. no. 1540012.
- [22] S. Zenasni, E. Kergosien, M. Roche, and M. Teisseire, "Spatial information extraction from short messages," *Expert Syst. Appl.*, vol. 95, pp. 351–367, Apr. 2018.
- [23] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han, "ClusType: Effective entity recognition and typing by relation phrase-based clustering," in *Proc. KDD*, 2015, pp. 995–1004.
- [24] M. B. Habib and M. Van Keulen, "TwitterNEED: A hybrid approach for named entity extraction and disambiguation for tweet," *Natural Lang. Eng.*, vol. 22, no. 3, pp. 423–456, May 2016.
- [25] M. Fossati, E. Dorigatti, and C. Giuliano, "N-ary relation extraction for simultaneous T-Box and A-Box knowledge base augmentation," *Semantic Web*, vol. 9, no. 4, pp. 413–439, Jun. 2018.
- [26] Z. Xu, X. Luo, S. Zhang, X. Wei, L. Mei, and C. Hu, "Mining temporal explicit and implicit semantic relations between entities using Web search engines," *Future Gener. Comput. Syst.*, vol. 37, pp. 468–477, Jul. 2014.
- [27] M. Conover, M. Hayes, S. Blackburn, P. Skomoroch, and S. Shah, "Pangloss: Fast entity linking in noisy text environments," in *Proc. 24th ACM SIGKDD*, 2018, pp. 168–176.
- [28] T. Boro, S. D. Dumitrescu, and R. Burtica, "NLP-Cube: End-to-end raw text processing with neural networks," in *Proc. CoNLL*, 2018, pp. 171–179.
- [29] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *Proc. 16th ICCL COLING*, vol. 1, 1996, pp. 1–6.
- [30] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proc. 27th ICCL*, 2018, pp. 2145–2158.
- [31] A. Goyal, V. Gupta, and M. Kumar, "Recent named entity recognition and classification techniques: A systematic review," *Comput. Sci. Rev.*, vol. 29, pp. 21–43, Aug. 2018.
- [32] R. Sharnagat, "Named entity recognition: A literature survey," in *Center For Indian Language Technology*. Bombay, India: Indian Institute of Technology, 2014.
- [33] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.
- [34] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, "A neural network framework for relation extraction: Learning entity semantic and relation pattern," *Knowl.-Based Syst.*, vol. 114, pp. 12–23, Dec. 2016.
- [35] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov. 2005.
- [36] R. Agerri and G. Rigau, "Robust multilingual Named Entity Recognition with shallow semi-supervised features," *Artif. Intell.*, vol. 238, pp. 63–82, Sep. 2016.
- [37] A. Passos, V. Kumar, and A. McCallum, "Lexicon infused phrase embeddings for named entity resolution," in *Proc. CoNLL*, 2014, p. 78.
- [38] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the Web: An experimental study," *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, Jun. 2005.
- [39] S. Zhang and N. Elhadad, "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts," *J. Biomed. Inform.*, vol. 46, no. 6, pp. 1088–1098, Dec. 2013.
- [40] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven pretraining of self-attention networks," 2019, *arXiv:1903.07785*. [Online]. Available: <http://arxiv.org/abs/1903.07785>
- [41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. NAACL*, 2019, pp. 4171–4186.
- [42] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. NAACL, Hum. Lang. Technol.*, vol. 1, 2018, pp. 2227–2237.
- [43] L. Liu, X. Ren, J. Shang, X. Gu, J. Peng, and J. Han, "Efficient contextualized representation: Language model pruning for sequence labeling," in *Proc. Conf. EMNLP, ACL*, 2018, pp. 1215–1225.
- [44] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.
- [45] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. 27th ICCL*, 2018, pp. 1638–1649.
- [46] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," 2018, *arXiv:1812.09449*. [Online]. Available: <http://arxiv.org/abs/1812.09449>
- [47] J. Plu, G. Rizzo, and R. Troncy, "Enhancing entity linking by combining NER models," in *Semantic Web Evaluation Challenge*, H. Sack, S. Dietze, A. Tordai, and C. Lange, Eds. Cham, Switzerland: Springer, 2016, pp. 17–32.
- [48] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 443–460, Feb. 2015.
- [49] C. Wang, X. Sun, H. Yu, and W. Zhang, "Entity disambiguation leveraging multi-perspective attention," *IEEE Access*, vol. 7, pp. 113963–113974, 2019.
- [50] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, "Entity disambiguation by knowledge and text jointly embedding," in *Proc. 20th SIGNLL*, 2016, pp. 260–269.

- [51] W. Zeng, J. Tang, X. Zhao, B. Ge, and W. Xiao, "Named entity disambiguation via probabilistic graphical model with embedding features," in *Neural Information Processing*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham, Switzerland: Springer, 2018, pp. 16–27.
- [52] R. Usbeck, A.-C. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both, "AGDISTIS-graph-based disambiguation of named entities using linked data," in *The Semantic Web—ISWC*. Cham, Switzerland: Springer, 2014, pp. 457–471.
- [53] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: A unified approach," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 231–244, Dec. 2014.
- [54] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 425–434.
- [55] F. Piccinno and P. Ferragina, "From TagME to WAT: A new entity annotator," in *Proc. 1st Int. Workshop Entity Recognit. Disambiguation*, 2014, pp. 55–62.
- [56] Z. Guo and D. Barbosa, "Robust named entity disambiguation with random walks," *Semantic Web*, vol. 9, no. 4, pp. 459–479, Jun. 2018.
- [57] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2015, pp. 238–243.
- [58] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1667–1676.
- [59] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "Pair-linking for collective entity disambiguation: Two could be better than all," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 7, pp. 1383–1396, Jul. 2019.
- [60] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, "Named entity disambiguation for noisy text," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2017, pp. 58–68.
- [61] S. Hakimov, H. T. Horst, S. Jebbara, M. Hartung, and P. Cimiano, "Combining textual and graph-based features for named entity disambiguation using undirected probabilistic graphical models," in *Knowledge Engineering and Knowledge Management*. Cham, Switzerland: Springer, 2016, pp. 288–302.
- [62] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [63] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proc. 24th Int. Conf. Artif. Intell. (IJCAI)*. New Orleans, LA, USA: AAAI Press, 2015, pp. 1333–1339.
- [64] M. Francis-Landau, G. Durrett, and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1256–1261.
- [65] F. Nie, Y. Cao, J. Wang, C.-Y. Lin, and R. Pan, "Mention and entity description co-attention for entity disambiguation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5908–5915.
- [66] O.-E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," in *Proc. Conf. EMNLP*, 2017, pp. 2619–2629.
- [67] Y. Eshel, N. Cohen, K. Radinsky, S. Markovitch, I. Yamada, and O. Levy, "Named entity disambiguation for noisy text," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2017, pp. 58–68.
- [68] O. Sevgili, A. Panchenko, and C. Biemann, "Improving neural entity disambiguation with graph embeddings," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Student Res. Workshop*, 2019, pp. 315–322.
- [69] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [70] G. Zhu and C. A. Iglesias, "Computing semantic similarity of concepts in knowledge graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 72–85, Jan. 2017.
- [71] D. B. Nguyen, M. Theobald, and G. Weikum, "J-NERD: Joint named entity recognition and disambiguation with rich linguistic features," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 215–229, Dec. 2016.
- [72] G. Luo, X. Huang, C.-Y. Lin, and Z. Nie, "Joint entity recognition and disambiguation," in *Proc. Conf. EMNLP*, vol. 2015, pp. 879–888.
- [73] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, "Analysis of named entity recognition and linking for tweets," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 32–49, Mar. 2015.
- [74] M. Van Erp, G. Rizzo, and R. Troncy, "Learning with the Web: Spotting named entities on the intersection of nerd and machine learning," in *Proc. CEUR Workshop*, 2013, pp. 27–30.
- [75] L. Canale, P. Lisena, and R. Troncy, "A novel ensemble method for named entity recognition and disambiguation based on neural network," in *Proc. ISWC*. Cham, Switzerland: Springer, 2018, pp. 91–107.
- [76] M. Liu, L. Chen, B. Liu, G. Zheng, and X. Zhang, "DBpedia-based entity linking via greedy search and adjusted Monte Carlo random walk," *SACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 1–34, Aug. 2017, doi: [10.1145/3086703](https://doi.org/10.1145/3086703).
- [77] A.-C. N. Ngomo and S. Auer, "LIMES—A time-efficient approach for large-scale link discovery on the Web of data," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, vol. 3, 2011, pp. 2312–2317.
- [78] A. Alhelbawy and R. Gaizauskas, "Graph ranking for collective named entity disambiguation," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 75–80.
- [79] W. Shen, J. Han, and J. Wang, "A probabilistic model for linking named entities in Web text with heterogeneous information networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data SIGMOD*, 2014, pp. 1199–1210, doi: [10.1145/2588555.2593676](https://doi.org/10.1145/2588555.2593676).
- [80] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang, "Learning entity representation for entity disambiguation," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 2. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 30–34. [Online]. Available: <https://www.aclweb.org/anthology/P13-2006>
- [81] R. Usbeck, A.-C. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both, "AGDISTIS-agnostic disambiguation of named entities using linked open data," in *Proc. 21st Eur. Conf. Artif. Intell. (ECAI)*, Amsterdam, The Netherlands, The Netherlands: IOS Press, 2014, pp. 1113–1114, doi: [10.3233/978-1-61499-419-0-1113](https://doi.org/10.3233/978-1-61499-419-0-1113).
- [82] B. Hachey, W. Radford, and J. R. Curran, "Graph-based named entity linking with wikipedia," in *Proc. 12th Int. Conf. Web Inf. Syst. Eng. (WISE)*. Berlin, Germany: Springer-Verlag, 2011, pp. 213–226. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2050963.2050980>
- [83] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani, "Learning relatedness measures for entity linking," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2013, pp. 139–148, doi: [10.1145/2505515.2505711](https://doi.org/10.1145/2505515.2505711).
- [84] G. Wu, Y. He, and X. Hu, "Entity linking: An issue to extract corresponding entity with knowledge base," *IEEE Access*, vol. 6, pp. 6220–6231, 2018.
- [85] D. B. Nguyen, J. Hoffart, M. Theobald, and G. Weikum, "Aida-light: High-throughput named-entity disambiguation," in *Proc. Workshop Linked Data Web Co-Located, 23rd Int. World Wide Web Conf. (WWW)*, Seoul, South Korea, vol. 1184, Apr. 2014, pp. 213–226. [Online]. Available: <http://dbis.eprints.uni-ulm.de/1206/>
- [86] Z. Zuo, G. Kasneci, T. Gruetze, and F. Naumann, "BEL: Bagging for entity linking," in *Proc. COLING 25th Int. Conf. Comput. Linguistics*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2075–2086. [Online]. Available: <https://www.aclweb.org/anthology/C14-1196>
- [87] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.* Portland, OR, USA: Association for Computational Linguistics, Jun. 2011, pp. 945–954. [Online]. Available: <https://www.aclweb.org/anthology/P11-1095>
- [88] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *Proc. Conf. EMNLP*, 2011, pp. 782–792.
- [89] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: A graph-based method," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2011, pp. 765–774, doi: [10.1145/2009916.2010019](https://doi.org/10.1145/2009916.2010019).
- [90] Z. Zheng, F. Li, M. Huang, and X. Zhu, "Learning to link entities with knowledge base," in *Proc. Hum. Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*. Los Angeles, CA, USA: Association for Computational Linguistics, Jun. 2010, pp. 483–491. [Online]. Available: <https://www.aclweb.org/anthology/N10-1072>
- [91] L. Ratniov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.* Portland, OR, USA: Association for Computational Linguistics, Jun. 2011, pp. 1375–1384. [Online]. Available: <http://www.aclweb.org/anthology/P11-1138>

- [92] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of Wikipedia entities in Web text," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2009, pp. 457–466, doi: 10.1145/1557019.1557073.
- [93] J. R. Raiman and O. M. Raiman, "DeepType: Multilingual entity linking by neural type system evolution," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.
- [94] I. Yamada and H. Shindo, "Pre-training of deep contextualized embeddings of words and entities for named entity disambiguation," 2019, *arXiv:1909.00426*. [Online]. Available: <http://arxiv.org/abs/1909.00426>
- [95] Z. Fang, Y. Cao, Q. Li, D. Zhang, Z. Zhang, and Y. Liu, "Joint entity linking with deep reinforcement learning," in *Proc. World Wide Web Conf.*, 2019, pp. 438–447.
- [96] A. Luo, S. Gao, and Y. Xu, "Deep semantic match model for entity linking using knowledge graph and text," *Procedia Comput. Sci.*, vol. 129, pp. 110–114, 2018.
- [97] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 1595–1604.
- [98] N. Kolitsas, O.-E. Ganea, and T. Hofmann, "End-to-end neural entity linking," in *Proc. 22nd Conf. Comput. Natural Lang. Learn.*, 2018, pp. 519–529.
- [99] Y. Yang, O. Irsoy, and K. S. Rahman, "Collective entity disambiguation with structured gradient tree boosting," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 777–786. [Online]. Available: <https://www.aclweb.org/anthology/N18-1071>
- [100] C. B. El Vaigh, F. Goasdoué, G. Gravier, and P. Sébillot, "Using knowledge base semantics in context-aware entity linking," in *Proc. ACM Symp. Document Eng. (DocEng)*, New York, NY, USA, 2019, pp. 8–1–8–10. [Online]. Available: <http://doi.acm.org/10.1145/3342558.3345393>
- [101] G. Rizzo, M. V. Erp, and R. Troncy, "Benchmarking the extraction and disambiguation of named entities on the semantic web," in *Proc. 9th LREC (LREC)*, 2014, pp. 4593–4600.
- [102] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva, "Analysis of named entity recognition and linking for tweets," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 32–49, Mar. 2015.
- [103] F. Corcoglioniti, M. Rospocher, and A. P. Aprosio, "A 2-phase frame-based knowledge extraction framework," in *Proc. 31st Annu. ACM Symp. Appl. Comput. (SAC)*, 2016, pp. 354–361.
- [104] A. Gangemi, "Semantic Web machine reading with FRED," *Semantic Web*, vol. 8, no. 6, pp. 873–893, 2017.
- [105] J. Bos, "Open-domain semantic parsing with boxer," in *Proc. 20th (NODALIDA)*, 2015, pp. 301–304.
- [106] S. Consoli and D. R. Recupero, "Using FRED for named entity resolution, linking and typing for knowledge base population," in *Semantic Web Evaluation Challenges*. Cham, Switzerland: Springer, 2015, pp. 40–50.
- [107] J. Plu, G. Rizzo, and R. Troncy, "A hybrid approach for entity recognition and linking," in *Semantic Web Evaluation Challenges*. Cham, Switzerland: Springer, 2015, pp. 28–39.
- [108] O.-E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann, "Probabilistic bag-of-hyperlinks model for entity linking," in *Proc. 25th Conf. WWW*, 2016, pp. 927–938.
- [109] Y. Cao, L. Hou, J. Li, and Z. Liu, "Neural collective entity linking," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 675–686. [Online]. Available: <https://www.aclweb.org/anthology/C18-1057>
- [110] P. H. Martins, Z. Marinho, and A. F. T. Martins, "Joint learning of named entity recognition and entity linking," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Student Res. Workshop*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 190–196. [Online]. Available: <https://www.aclweb.org/anthology/P19-2026>
- [111] M. van Erp, P. Mendes, H. Paulheim, F. Ilievski, J. Plu, G. Rizzo, and J. Waitelonis, "Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job," in *Proc. LREC*, May 2016, pp. 4373–4379. [Online]. Available: <https://www.aclweb.org/anthology/L16-1693>
- [112] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A.-L. Minard, A. Palmero Aprosio, G. Rigau, M. Rospocher, and R. Segers, "NewsReader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news," *Knowl.-Based Syst.*, vol. 110, pp. 60–85, Oct. 2016.
- [113] Y. Chen, Y. Tan, Q. Wu, and W. Wang, "TGCEL: A chinese entity linking method based on topic relation graph," in *Proc. 6th Int. Conf. Comput. Sci. New Technol. (ICCSNT)*, 2017, pp. 226–230.
- [114] R. Weegar, A. P. Ramírez, H. Dalianis, K. G. Galleitebeita, A. C. Rubio, and M. Oronoz Anchochoqui, "Ensembles for clinical entity extraction," *Sociedad Española para el Procesamiento del Lenguaje Natural*, vol. 60, 2018.
- [115] S. S. Sazali, N. A. Rahman, and Z. A. Bakar, "Information extraction: Evaluating named entity recognition from classical malay documents," in *Proc. 3rd Int. Conf. Inf. Retr. Knowl. Manage. (CAMP)*, 2016, pp. 48–53.
- [116] A. Marginean, A. Groza, R. R. Slavescu, and I. A. Letia, "Romanian2SPARQL: A grammatical framework approach for querying linked data in romanian," in *Proc. Int. Conf. Develop. Appl. Syst. (DAS)*, 2014, pp. 204–209.
- [117] A. Kirillovich, O. Nevzorova, E. Gimadiev, and N. Loukachevitch, "RuThes cloud: Towards a multilevel linguistic linked open data resource for Russian," in *Proc. Int. Conf. Knowl. Eng. Semantic Web*. Cham, Switzerland: Springer, 2017, pp. 38–52.
- [118] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *Expert Syst. Appl.*, vol. 114, pp. 34–45, Dec. 2018.
- [119] T. Kenter, L. Jones, and D. Hewlett, "Byte-level machine reading across morphologically varied languages," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5829–5827.
- [120] A. Weichselbraun, P. Kuntschik, and A. M. Braoveanu, "Mining and leveraging background knowledge for improving named entity linking," in *Proc. 8th Int. Conf. Web Intell., Mining Semantics*, 2018, p. 27.
- [121] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5464–5472.
- [122] S. Upadhyay, N. Gupta, and D. Roth, "Joint multilingual supervision for cross-lingual entity linking," in *Proc. Conf. EMNLP*, 2018, pp. 2486–2495.
- [123] J. Xie, Z. Yang, G. Neubig, N. A. Smith, and J. Carbonell, "Neural cross-lingual named entity recognition with minimal resources," in *Proc. Conf. EMNLP*, 2018, pp. 369–379.
- [124] S. Rijhwani, J. Xie, G. Neubig, and J. Carbonell, "Zero-shot neural transfer for cross-lingual entity linking," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6924–6931.
- [125] R. Usbeck, "GERBIL: General entity annotator benchmarking framework," in *Proc. 24th Conf. WWW*, 2015, pp. 1133–1143.
- [126] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N<sup>3</sup>—A collection of datasets for named entity recognition and disambiguation in the NLP interchange format," in *Proc. LREC*, 2014, pp. 3529–3533.
- [127] F. J. Azanzi and G. Camara, "Knowledge extraction from source code based on hidden Markov model: Application to EPICAM," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2017, pp. 1478–1485.
- [128] G. R. Rao, V. Balaran, and B. Vishnuvardhan, "Knowledge extraction from software engineering repositories," in *Proc. Intell. Syst. Conf. (IntelliSys)*, 2017, pp. 366–372.
- [129] A. Jiomekong and G. Camara, "An approach for knowledge extraction from source code (KNESC) of typed programming languages," in *Proc. World Conf. Inf. Syst. Technol.* Cham, Switzerland: Springer, 2018, pp. 122–131.
- [130] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, Feb. 2018.
- [131] B. Y. Lin and W. Lu, "Neural adaptation layers for cross-domain named entity recognition," in *Proc. Conf. EMNLP*, 2018, pp. 2012–2022.
- [132] J. Xu, H. He, X. Sun, X. Ren, and S. Li, "Cross-domain and semisupervised named entity recognition in chinese social media: A unified model," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 11, pp. 2142–2152, Nov. 2018.
- [133] C. Jia, X. Liang, and Y. Zhang, "Cross-domain NER using cross-domain language modeling," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2464–2474.
- [134] D. Maynard, I. Roberts, M. A. Greenwood, D. Rout, and K. Bontcheva, "A framework for real-time semantic social media analysis," *J. Web Semantics*, vol. 44, pp. 75–88, May 2017.
- [135] A. E. Cano, "Making sense of microposts:(# microposts2014) named entity extraction & linking challenge," in *Proc. CEUR Workshop*, vol. 1141, 2014, pp. 54–60.

- [136] M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo, "Gerbil—benchmarking named entity recognition and linking consistently," *Semantic Web*, vol. 9, no. 5, pp. 605–625, 2018.
- [137] K. Jha, M. Röder, and A.-C. N. Ngomo, "All that glitters is not gold—rule-based curation of reference datasets for named entity recognition and entity linking," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2017, pp. 305–320.
- [138] R. Lima, B. Espinasse, and F. Freitas, "Relation extraction from texts with symbolic rules induced by inductive logic programming," in *Proc. IEEE 27th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2015, pp. 194–201.
- [139] G. Gorrell, J. Petrak, and K. Bontcheva, "Using @Twitter conventions to improve #LOD-based named entity disambiguation," in *The Semantic Web Latest Advances and New Domains*. Cham, Switzerland: Springer, 2015, pp. 171–186.
- [140] H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva, "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics," *PLoS Comput. Biol.*, vol. 9, no. 2, Feb. 2013, Art. no. e1002854.
- [141] R. Speck and A.-C. N. Ngomo, "Ensemble learning for named entity recognition," in *Proc. Int. Semantic web Conf.* Cham, Switzerland: Springer, 2014, pp. 519–534.
- [142] F. Boudin, "PKE: An open source python-based keyphrase extraction toolkit," in *Proc. COLING*, 2016, pp. 69–73.
- [143] A. Gangemi, "A comparison of knowledge extraction tools for the semantic Web," in *Proc. Extended Semantic Web Conf.* Berlin, Germany: Springer, 2013, pp. 351–366.
- [144] D. Damjanovic and K. Bontcheva, "Named entity disambiguation using linked data," in *Proc. 9th Extended Semantic Web Conf.*, 2012, pp. 231–240.
- [145] R. S. João, P. Fafalios, and S. Dietze, "Same but different: Distant supervision for predicting and understanding entity linking difficulty," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, 2019, pp. 1019–1026.
- [146] T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann, "Coheel: Coherent and efficient named entity linking through random walks," *J. Web Semantics*, vols. 37–38, pp. 75–89, Mar. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570826816000172>
- [147] C. Rao and V. N. Gudivada, *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*, vol. 38. Amsterdam, The Netherlands: Elsevier, 2018.
- [148] A. Johnson, P. Karanasou, J. Gaspers, and D. Klakow, "Cross-lingual transfer learning for japanese named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2019, pp. 182–189.
- [149] J. T. Zhou, H. Zhang, D. Jin, H. Zhu, M. Fang, R. S. M. Goh, and K. Kwok, "Dual adversarial neural transfer for low-resource named entity recognition," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3461–3471.
- [150] S. Hu, Z. Tan, W. Zeng, B. Ge, and W. Xiao, "Entity linking via symmetrical attention-based neural network and entity structural features," *Symmetry*, vol. 11, no. 4, p. 453, Apr. 2019.
- [151] A. G. Nuzzolese, A. L. Gentile, V. Presutti, A. Gangemi, R. Meusel, and H. Paulheim, "The second open knowledge extraction challenge," in *Semantic Web Challenges*, H. Sack, S. Dietze, A. Tordai, and C. Lange, Eds. Cham, Switzerland: Springer, 2016, pp. 3–16.



**TAREQ AL-MOSLMI** received the B.Sc. degree (Hons.) in computer science from the University of Science and Technology, Yemen, in 2009, the master's degree in computer science and the Ph.D. degree from the Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2014 and 2018, respectively. He is currently a Postdoctoral Fellow with the University of Bergen, Norway. His research interests fall under natural-language processing, machine learning, semantic and knowledge graphs, text and web mining, and sentiment analysis.



**MARC GALLOFRÉ OCAÑA** received the B.Sc. degree informatics engineering and the M.Sc. degree in innovation and research in informatics from the Polytechnic University of Catalonia-BarcelonaTech. He is currently pursuing the Ph.D. degree with the University of Bergen. His current research includes knowledge graphs and big-data software architectures to support journalistic knowledge platforms. His past researches fall under information systems strategic planning and informational analysis of university rankings. He had been previously working as a business intelligence consultant in different sectors like sports, finance, and emergency services.



**ANDREAS L. OPDAHL** received the Ph.D. degree from the Norwegian University of Science and Technology, in 1992. He is currently a Professor of information systems development with the University of Bergen, Norway. His research interests include ontologies and knowledge graphs, enterprise, IS modeling, and safety and security requirements. He is the author, a coauthor or the Co-Editor of more than a hundred peer-reviewed research articles that have been cited more than 3700 times. He is also a member of IFIP WG5.8 on enterprise interoperability and WG8.1 on design and evaluation of information systems. He serves regularly as a Reviewer for premier international journals and on the program committees and as an Organizer of the most renowned international conferences and workshops in his fields of interest.



**CSABA VERES** received the Ph.D. degree in cognitive science from the University of Arizona, but migrated to computer science, from 1999 to 2000. He is currently an Associate Professor with the University of Bergen, Norway. Since then, he has worked as a Computer Scientist. His main interest has always been semantics, and he carried this into computer science. His areas of expertise include NLP, machine learning, and semantic web. He has experience as an Academic and Practitioner. He has published original research articles, consulted with technology startups. He has popular linked data apps on the Apple store, called MapXplore and AuotoMind.