Unsupervised Approaches for Textual Semantic Annotation, A Survey

XIAOFENG LIAO and ZHIMING ZHAO, University of Amsterdam, Netherlands

Semantic annotation is a crucial part of achieving the vision of the Semantic Web and has long been a research topic among various communities. The most challenging problem in reaching the Semantic Web’s real potential is the gap between a large amount of unlabeled existing/new data and the limited annotation capability available. To resolve this problem, numerous works have been carried out to increase the degree of automation of semantic annotation from manual to semi-automatic to fully automatic. The richness of these works has been well-investigated by numerous surveys focusing on different aspects of the problem. However, a comprehensive survey targeting unsupervised approaches for semantic annotation is still missing and is urgently needed. To better understand the state-of-the-art of semantic annotation in the textual domain adopting unsupervised approaches, this article investigates existing literature and presents a survey to answer three research questions: (1) To what extent can semantic annotation be performed in a fully automatic manner by using an unsupervised way? (2) What kind of unsupervised approaches for semantic annotation already exist in literature? (3) What characteristics and relationships do these approaches have?

In contrast to existing surveys, this article helps the reader get an insight into the state-of-art of semantic annotation using unsupervised approaches. While examining the literature, this article also addresses the inconsistency in the terminology used in the literature to describe the various semantic annotation tools’ degree of automation and provides more consistent terminology. Based on this, a uniform summary of the degree of automation of the many semantic annotation tools that were previously investigated can now be presented.

CCS Concepts: • General and reference → Surveys and overviews; • Information systems → Entity relationship models, Ontologies, Information extraction; • Theory of computation → Unsupervised learning and clustering; • Computing methodologies → Unsupervised learning; Semantic networks;

Additional Key Words and Phrases: Semantic annotation, unsupervised, information extraction, entity recognition, relation extraction, entity linking, machine learning

ACM Reference format:
https://doi.org/10.1145/3324473

This research is supported by the European Union’s Horizon 2020 research and innovation program under Grants No. 654182 (ENVRIPLUS), No. 824068 (ENVRI-FAIR), No. 825134 (ARTICONF), and No. 676247 (VRE4EIC). The first author also acknowledges the support from National Natural Science Foundation of China under Grant No. 61762062.

Authors’ addresses: X. Liao and Z. Zhao (corresponding author), Informatics Institute, University of Amsterdam, Science Park 904, 1098XH Amsterdam, Netherlands; emails: {X.Liao, z.zhao}@uva.nl.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2019 Copyright held by the owner/author(s).
0360-0300/2019/08-ART66
https://doi.org/10.1145/3324473


1 INTRODUCTION

1.1 Definition and Scope

Tim Berners-Lee described the Semantic Web in the early 2000s as “an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” (Berners-Lee et al. 2001). It was the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications (Guha et al. 2003).

The fundamental foundations of the Semantic Web are ontologies and annotations. Ontologies, which are normally represented in the Resource Description Framework (RDF) or Web Ontology Language (OWL), describe formally shared conceptualizations of a domain (e.g., singer, album). Annotations allow binding ontology-based descriptions to existing Web resources and are normally described in RDF. In summary, HTML describes documents and the links between them, while RDF, RDF Schema, and OWL, by contrast, can describe arbitrary things.

A few years later, Tim Berners-Lee again proposed the Linked Data principles (Bizer et al. 2011) leading to the publication and growth of linked open datasets weaving a Web of Linked Data.

The concept of the Web of Linked Data has appeared as a mechanism for making all RDF data available using the HTTP protocol, as is done with HTML documents.

Though these concepts have different emphasis, they are to some extent just different facets of a specific evolution of the Web to make the Web more machine-friendly and to support more automation (Gandon 2018).

Semantic annotation is the supporting effort for all these concepts. It involves tagging of data with concepts (i.e., ontology classes) so that data becomes meaningful, hence an increase interoperability (Ghijsen et al. 2013). Annotating data can help in providing better search facilities since it helps users to search for information not only based on the traditional keyword-based search but also using well-defined general concepts that describe the domain of their information need (Giannopoulos et al. 2010).

The widely adopted definition of Semantic Annotation is the process of mapping between ontologies (classes, concepts, properties, and relations) and their instances in a document, with the purpose of facilitating knowledge enrichment of the document, and is intended primarily for use by machines, empowering users to search across resources for entities and relations based on meaning instead of syntax (Bontcheva and Cunningham 2011; Kiryakov et al. 2004; Oliveira and Rocha 2013; Reeve and Han 2005, 2006; Tamper et al. 2016; Tang et al. 2008; Uren et al. 2006).

Semantic annotation can be viewed from the Information Extraction perspective. Information extraction using Natural Language Processing technologies to extract information from unstructured textual input and produce a structured textual output, further may populate a structured data or knowledge base. The information that can be of interest includes named entities, relations connecting named entities, and even a more complex type of information, the event, which can be seen as a group of relations grounded in time.

When the named entities are not merely a flat structure but linked to a corresponding entity in an ontology, this is known as semantic annotation or named entity linking (NEL). Semantic annotation is much more potent than flat NE recognition, because it enables inferences and generalizations to be made, as the linking of information provides access to knowledge not explicit in the text. In the context of semantic annotation, the information extraction task is often referred to as Ontology-Based Information Extraction (OBIE) or Ontology Guided Information Extraction. Closely associated with this is the process of ontology learning and population (Maynard et al. 2016).

It is notable that according to the aforementioned definition, semantic annotation is not limited in associating entities/named entities and relations to a knowledge base. Semantic annotation
might also apply to classes, concepts, and properties that are not discrete named entities. The term “Named Entity,” now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim 1996). At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense-related activities is extracted from the unstructured text, such as newspaper articles (Nadeau and Sekine 2007).

This survey mainly focuses on the information extraction component of semantic annotation, where automation is often applied, specifically on fully automatic or unsupervised ontology-based information extraction tools and algorithms.

Semantic annotation may occur in not only the text domain but also in other media such as image or video. This article, however, focuses explicitly on unstructured text in natural language. Currently, the most promising results are achieved via manual/supervised approaches or hybridization of the two constrained in some settings with varying amounts of additional human expertise. It is still desirable, however, when facing the sheer quantity of existing and new data needing to be annotated, that researchers try to make the semantic annotation process more autonomous and general, and thus reduce the need for human intervention as much as possible. Better unsupervised semantic annotation is needed for the following reasons including but not limited to

— People who publish web pages and documents do not voluntarily annotate their content.
— Manual annotation or supervised annotation is exhaustively expensive and prone to subjective interpretation or error.
— Large quantities of legacy documents need to be annotated.
— The huge amount of new documents being generated outpaces the capability that manual or supervised annotation can deal with.

1.2 Research Questions

The semantic annotation has long been studied. This can be seen in the richness of existing survey and review papers relating to semantic annotation tools and algorithms, which will become clear in Table A.3. However, examination of these surveys reveals that the degree of automation was only studied in some of the papers, and was never the dedicated topic of any paper. The exception, (Tang et al. 2008), was a survey paper devoted to automatic semantic annotation, but their focus was only on semi-supervised algorithms.

Based on this observation, a comprehensive investigation targeting the status of fully automatic semantic annotation is indeed still needed. This drives us to find answers to the following research questions through this survey:

— Question 1: To what extent can semantic annotation be completed in a fully automatic or unsupervised way?
Solution: Since semantic annotation is a procedure consisting of several steps (pre-processing, named entity recognition, relation recognition, and so on) and two parties (a document and an ontology), with many works targeting only part of the whole procedure, we answer this question by first presenting a general flow of semantic annotation which consists of a number of detailed steps, and investigate through existing literature which steps are fully automated or unsupervised.

— Question 2: What kind of fully automatic or unsupervised approaches for semantic annotation exist in the literature?
Solution: After answering Question 1, it is natural to summarize and categorize the methods adopted in the literature for performing semantic annotation.
Question 3: What characteristics and relationships do these approaches have?

Solution: After categorizing into different groups based on methodology, we summarize and categorize the methods observed from literature and analyze their properties.

By investigating and answering the research questions listed above, this survey helps the reader to understand what fully automatic or unsupervised approaches exist for doing semantic annotation, and what are their characteristics and what are their pros and cons compared to the tools that use supervised or semi-supervised approaches.

1.3 Survey Structure

In Section 1.1, We describe the widely adopted definition of semantic annotation and its relation with some other terms, such as information extraction and relation extraction. Section 1.2 raises the three research questions that this survey addresses. This subsection presents a visual depiction (Figure 1) of how this article is organized and how the results relate to the research questions raised in the previous section.

Section 2 describes the method we used to do the literature search. It consists of two main parts: the first part describes how we build query strings to identify 12 related survey papers, while the second part describes how we build query strings to identify related literature further. Complementary manual searches also included where helpful.

In Section 3, we summarize the contribution of the 12 survey papers identified in Section 2 and demonstrate the uniqueness of the point of view of this survey. We also address the mess of terms used to describe the degree of automation in literature and propose a uniform scale for describing the degree of automation.

We answer research question 1 by firstly establishing a sub-tasks layer stack under the scope of semantic annotation in Section 4.1.1. Then we investigate the degree of automation of notable works in four critical sub-tasks in Section 4.1.2, Section 4.1.3, Section 4.1.4, and Section 4.1.5, respectively. In Section 4.2, we answer the research questions 2 and 3 by summarizing the approaches those selected semantic annotation tools used and present a classification. We also analyze the characteristics of each approach. Finally, in Section 5 we conclude our work and present several trends identified in the process of constructing our survey.

2 SURVEY METHODOLOGY

2.1 Search Strategy

Semantic annotation has been studied and developed widely for decades. In recent years, we have seen a wealth of research dedicated to semantic annotation focusing on various aspects of the problem. There also have many existing survey papers from various perspectives. We believe
understanding these existing survey papers would be helpful to begin a new comprehensive survey regarding unsupervised approaches.

2.1.1 Survey Paper Identification. In this subsection, we briefly describe the method we used to search for existing survey papers, which is illustrated in black in Figure 2. We used three tools to do the literature search: Web of Science, Scopus, and Google Scholar. We built Query 1 and submitted it to these three tools, respectively, then aggregated the results. As each of these three tools has different coverage, by querying all of them and aggregating the results together, we were able to increase the coverage of our literature search.

The search strings for Query 1 submitted to Scopus, Web of Science, and Google Scholar respectively, along with the count of results for each search and the final count of documents retained after examination are listed in Table A.1.

2.1.2 Further Literature Search. After identifying and further examining the existing survey papers, it was revealed that the topic of automation is only partially discussed. Thus, a further literature search is needed to increase our own survey’s coverage.

This further literature search was conducted using Query 2. The search strings for Query 2, along with the count of results for each search and the final count of documents retained after examination are listed in Table A.2.

As a result, a final list of 12 survey papers was aggregated from the Survey Paper Identification phase, which is listed in Table A.3. A final literature set of 367 papers was determined in the Further Literature Search phase. For detailed search methodology, please refer to Appendix A.

---

1Both Query 1 and Query 2 were last checked in November, 2018.
Table 1. Scales Used in the Five Survey Papers to Distinguish the Degree of Automation

<table>
<thead>
<tr>
<th>Survey paper</th>
<th>Terms used to describe the degree of automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fähndrich et al. (2015)</td>
<td>“None”, “Semi”, “Collection”, or “Full”</td>
</tr>
</tbody>
</table>

The complete process of our literature search is illustrated in Figure 2. The workflow for Q1 and Q2 is shown in black and blue, respectively, and the complementary manual search is in red.

3 COMPARATIVE STUDY OF RELATED WORKS

3.1 Comparison of Surveys

In this subsection, we briefly summarize and highlight the significant contribution of each of the existing survey papers listed in Table A.3 and contrast them with this survey.

Several observations can be drawn directly from the summarization in Tables A.3 and A.4:

1. There is a wealth of research focusing on different aspects of semantic annotation. However, only a few surveys were partially concerned with the topic of degree of automation. Of the 12 survey papers in Table A.3, five of them investigated the topic of the degree of automation. Liao et al. (2015) used one column of their comparison table to describe how semantic annotations were added to the annotated elements, either manual, semi-automatic or automatic. Oliveira and Rocha (2013) also compared different annotation methods regarding the level of automation, with the scale from manual to fully automated. An observation they concluded was that fully automatic systems were still a challenge to make and operate. Tang et al. (2008) focused on semi-supervised machine learning techniques. In none of these surveys was the degree of automation a primary concern.

2. Inconsistencies exist in the scales used to distinguish the degree of automation. Another observation that can be drawn is that different scale terms were used to describe the degree of automation. Table 1 illustrates the mess of terms used.

Liao et al. (2015) used three degrees, “Manual,” “Semi-automatic,” or “Automatic,” to distinguish how semantic annotations are being added to the annotated elements. In Oliveira and Rocha (2013), the terms describing the level of automation of the method were a bit unclear. It stated that the level of automation goes from manual to fully automated. In their summary table, three degrees appeared, “manual,” “semi-automatic,” and “automatic.” Based on their statement, “Systems known are mainly semi-automatic approaches while fully automatic systems are still a challenge,” we speculate with surety that their degree of automation consists of “manual,” “semi-automatic,” “automatic,” and “fully automatic.” As Tang et al. (2008) mainly focused on “semi-automatic” methods, they did not provide a scale to measure the degree of automation. Among the many annotators they investigated, they simply regarded two methods “KNOWITALL” and “TEXTRUNNER” as unsupervised, which is supposed to have higher automation than the semi-automatic degree. We speculate with surety that their degree of automation consists of two scales: “semi-automatic” and “unsupervised.” Uren et al. (2006) provided a table for comparison of automation of annotation tools; among them, six tools were classified as using “unsupervised learning,” and one that was “supervised and unsupervised.” The scale they employed consists of three scales: “No,” “Supervised,” and “Unsupervised.” Fähndrich et al. (2015) provided an overview
of available tools utilizing different properties, one of the properties being the level of automation, which was distinguished into four categories: “None,” “Semi,” “Collection,” or “Full.” They described “Semi” as the ability to automatically perform some task with the constraint that there is still the requirement to supervise the process. Since they believe the collection of information is a time-consuming task which needs to be automated, they specifically defined “Collection” as a degree of automation to describe the ability to Collection information automatically.

The confusion caused by using these inconsistent terms can easily be shown by the following examples:

—The MnM tool (Vargas-Vera et al. 2002) was regarded as semi-automatic in Fähndrich et al. (2015) and Oliveira and Rocha (2013), while regarded as automatic in Liao et al. (2015).

—In AeroDAML (Kogut and Holmes 2001), annotation behavior done using the drag-and-drop interface is defined as semi-automatic. However, in Cerno (Kiyavitskaya et al. 2009), annotation by typing or in a drag-and-drop manner is regarded as manual, while using wrappers and information extraction components are regarded as semi-automatic.

3. The term “Automatic” was widely and loosely used. The term “automatic” was widely adopted to describe the degree of automation. However, “automatic” is very general and loosely defined, and any application of NLP techniques to assist the annotation was referred to as "automatic.” These observations indicate that a uniform scale is strongly needed to measure the degree of automation to avoid ambiguity. We use Table 2 to help remove disambiguation and make the description of the degree of automation more clear.

—When machine learning techniques were used in the annotation, “unsupervised” or “semi-supervised” should be used.

—“Unsupervised” means that no labeled training sample is needed to train the system.

—In contrast, “semi-supervised” means labeled training data is needed.

—In the case of no machine learning algorithm used, “manual,” “semi-automatic,” or “fully automatic” should be used.

—“Fully automatic” means that no human intervention is required in the whole process, except for some initial input.

—“Semi-automatic” means that human intervention is required in some phases.

—“Manual” indicates that the whole process requires human intervention.

—Simply “automatic” is discarded because its meaning is too generous. Many works use “automatic” even when there is a need for partial human intervention in their complete process.

3.2 A Uniform Summary of the Degree of Automation

By using the terms in Table 2, a uniform summary regarding some notable tools from Table A.4 is presented in Table A.5. Note that we classify some notable tools, such as PANKOW and Armadillo, which were previously regarded as unsupervised, as fully automatic due to the fact that they do not use machine learning techniques to function.
PANKOW calls itself unsupervised as it does not rely on any training data annotated by hand. However, it is pattern-based in the sense that it makes use of linguistically motivated regular expressions to identify instance-concept relations in text, which requires human expertise to design matching patterns. Also, the design of the pattern does not fall into an unsupervised machine learning paradigm where optimal parameters of a model are achieved via a process approaching an objective function without human-annotated training data. Similarly, Armadillo does not fall into the unsupervised machine learning paradigm as it heavily relies on human expertise to build wrappers to extract information from highly regular web pages.

Here, several conclusions can be drawn from Table A.5 as quick answers to the three research questions. To be noted, these conclusions are generated from the 12 present survey papers, which are identified through our search strategy. A further detailed study is presented in the next section.

— There are only several tools that are fully automatic or unsupervised. They are DIPRE, Snowball, AeroDAML, SemTag, PANKOW, Armadillo, KnowItAll, Textrunner, and StatSnowball, which are shown in bold.

— There are three main approaches adopted by such tools to improve the degree of automation: bootstrapping approach, Wrapper induction, and exploiting redundancy.

— Bootstrapping has been an attractive alternative in automatic text processing, and can respectively exploit structure traits or pattern traits. Bootstrapping methods significantly reduce the number of training examples by iteratively discovering extraction patterns from new observations and identifying entity relations with a small number of seeds, either target relation tuples or general extraction templates. Bootstrapping methods do not perform open information extraction, which can identify various types of relations without requiring pre-specifications.

— Wrappers strongly rely on document structure and work best for collections of rigidly structured web pages. The most significant advantage of wrappers is that they need a small number of training data (Kiyavitskaya et al. 2009).

— Only a few tools utilize machine learning techniques, mostly by utilizing bootstrapping or pattern-based approaches.

### 4 RESULTS

In addition to the quick conclusions drawn in the previous section, the literature set identified by our Further Literature Search phase is explored and detailed answers are presented in this section.

#### 4.1 Results Regarding Research Question 1

To answer research question 1, that is to what extent semantic annotation can be completed in a fully automatic or unsupervised way, it is first necessary to establish the sub-tasks that together constitute the complex task of semantic annotation. In Section 4.1.1, common sub-tasks are aggregated from the literature set. As Figure 3(a) indicates, the Entity Related and Relation Related tasks are the two most studied sub-tasks in the scope of semantic annotation. Then, four key sub-tasks, i.e., Named Entity Recognition, Relation Extraction, Entity Linking, and Ontology Development are discussed, respectively, in Sections 4.1.2, 4.1.3, 4.1.4, and 4.1.5, in terms of degree of automation and performance.

#### 4.1.1 Sub-Task Establishment

Common sub-tasks of semantic annotation observed in the literature set include but are not limited to the following:

— At the entity level: named entity extraction, named entity recognition, named entity disambiguation, named entity identification, and named entity linking.
At the relation level, relation extraction, relation recognition, relation discovery, and relation categorization.

— Tasks involving ontology development, such as ontology learning and ontology population, ontology refinement.

— Other tasks, including word sense disambiguation, and some shallow natural language parsing tasks such as tokenization, sentence splitting, and POS Tagging.

The proportion of different sub-tasks is presented in Figure 3(a). Note, this division of sub-tasks is not completely precise, as the classification is based solely on the title and abstract of surveyed literature. The majority of sub-tasks are “entity” and “relation” oriented, and thus a further division of entity-related tasks and relation-related task are presented in Figures 3(b) and 3(c), respectively. Note, that some works perform two or more sub-tasks, such as extraction and classification, together.

— At the “Entity” level, the most common tasks are “Entity Recognition” and “Entity Extraction.”

— At the “Relation” level, the most used terms are “Relation Extraction,” “Relation Discovery,” and “Relation Identification.” Most of these tasks overlap with the concerns of the field of Information Extraction.

Information extraction is the process of extracting information from unstructured free text and turning it into structured data, and further may populate a structured data or knowledge base. Several types of information that can be of interest include named entities (NEs), relations connecting named entities together, and even a more complex type of information, the event, which can be seen as a group of relations grounded in time (Maynard et al. 2016). Information extraction typically consists of a sequence of tasks, comprising (1) linguistic pre-processing; (2) named entity recognition; and (3) relation and/or event extraction.

When the named entities are not simply a flat structure but linked to a corresponding entity in an ontology, this is known as semantic annotation or named entity linking (NEL). Semantic annotation is much more potent than flat NE recognition, because it enables inferences and generalizations to be made, as the linking of information provides access to knowledge not explicit in the text. In the context of semantic annotation, the information extraction task is often referred to as Ontology-Based Information Extraction (OBIE) or Ontology Guided Information Extraction. Closely associated with this is the process of ontology learning and population (Maynard et al. 2016).

Since the names of some tasks at the entity and relation level are close and similar, which may easily confuse, it is good to make their scope clear. A list of the objectives of each closely related task at the entity and relation level is aggregated and presented below. Examples are also presented for a better understanding of each task.

— Named Entity Recognition (NER) (a.k.a. Named Entity Extraction), deals with the locating of entity mentions (person, location, etc.) in text and their classification into predefined
semantic types. This involves two sub-tasks, among which the first task is about the segmentation of token sequences to obtain mentioned boundaries. The second tasks can be referred to as Named Entity Classification (NEC), and mapping relevant token spans from the first task to predefined classes or types (Finkel et al. 2005; Maynard et al. 2016).

NER is often used to mean the combination of the two tasks. In contrast, Maynard et al. (2016) and Nadeau and Sekine (2007) use NERC for the combined task and NER for only the recognition element.

For example, NER on the text “Einstein won the Nobel Prize identifies,” finds the mentions “Einstein” and “Nobel Prize” and marks them as being of “person” and “misc” type, respectively (Dutta and Weikum 2015).
—Named Entity Linking (also referred to as record linkage or entity resolution or entity disambiguation), is a further task regarding named entities, which is to recognize if a named entity mention in a text corresponds to any NEs in a reference knowledge base, such as Wikipedia, DBpedia, or YAGO. The knowledge base, however, may or may not contain the entity. It typically requires annotating a potentially ambiguous entity mention in a document (e.g., Paris) with a link to a canonical identifier describing a unique entity in a database or an ontology (e.g., http://dbpedia.org/resource/Paris) (Hoffart et al. 2011; Lin et al. 2012; Maynard et al. 2016; Rao et al. 2013; Shen et al. 2015).

NEL approaches typically include a candidate selection phase, which identifies all candidate knowledge base entries for a given entity mention in text. This is followed by a Named Entity Disambiguation (or Entity Resolution) phase, which links named entity mentions onto their corresponding entries in a knowledge base (Han and Zhao 2009; Hoffart et al. 2011; Li et al. 2013b; Maynard et al. 2016; Rao et al. 2013).
—Relation extraction is about being able to identify and classify various semantic relations between named entities mentioned in the text, such as organization-location or person-affiliation relations. It can be seen as a combination of relation identification and relation classification.

It builds on the task of NERC discussed in the previous paragraphs. The focus is often to extract binary relations between named entities. Typical relation types include birthdate (PER, DATE) and founder-of (PER, ORG), with examples for relations being birthdate (John Smith, 1985-01-01) or founder-of (Bill Gates, Microsoft) (Bollegala et al. 2010; Maynard et al. 2016; Min et al. 2012).

It can be subdivided into three tasks: relation argument identification (finding the boundaries of the arguments), relation argument classification (assigning types to the arguments), and relation classification (assigning a type to the relation) (Augenstein et al. 2015). The first two tasks are generally approached using NERC.
—Ontology development is a critical player in semantic annotation; we hereby take the definition of some ontology development related tasks from Maynard et al. (2016). We generally refer to them as ontology development, which can be seen consists of three related components: learning, population, and refinement. Ontology learning (or generation) denotes the task of creating an ontology from scratch, and mainly concerns the task of defining the concepts and generating relevant relations between them. Ontology population consists of adding instances to an existing ontology structure (as created by the ontology learning task, for instance). Ontology refinement involves adding, deleting, or changing new concepts, relations, and/or instances in an existing ontology. Ontology learning may also be used to denote all three tasks, in particular where the tasks of learning and population are performed via a single methodology.
We organize the main sub-tasks described above in a layer cake according to the logical order the sub-tasks happen during a typical semantic annotation procedure. This layer is shown in Figure 4. This layer cake is referred to as the **semantic annotation layer cake**.

### 4.1.2 Named Entity Recognition

Three semantic annotation tools doing entity recognition task in a fully automatic or unsupervised manner, namely, SemTag, Armadillo, and KnowItAll, are selected from Table A.5 to compare with works identified in the literature set. A comparison is presented to see the advancement in the field of named entity recognition by the unsupervised approach.

We examine the human input required in some notable works and also present the performance reported by the authors in Table A.6. Due to the variance of the task scenarios, as well as the tailored evaluation procedures and metrics adopted, it is not feasible to evaluate them uniformly. For this reason, we merely summarize and present author-reported results. The selection criteria for these works are mainly based on their impact, measured by metrics such as citation count and the impact factor of the conference or journal in which the work is published. We limit ourselves to those works that report a complete system, instead of those merely reporting prototype or algorithms. The time span covered is from 2001 to 2018.

In a task of mining Computer Science (CS) Department websites to extract staff data, Armadillo (Dingli et al. 2003b) limited the requirement of manual annotation to only a list of CS websites. The patterns induced from such examples, in their result, achieved high precision and high recall. The key feature of their task, ensuring the satisfying results, was the “redundancy” of the targeted info on the Web. They experimented on several websites of the Computer Science Department in UK Universities to find people’s names. Their results reached high precision: 0.94, recall: 0.37, and F-Measure: 0.51.

SemTag (Dill et al. 2003b) tags huge numbers of pages (264 million), with terms from a standard ontology, and generated approximately 434 million automatically disambiguated semantic tags. It was the most massive scale semantic tagging effort to the year they published, 2003. The human input required was the knowledge base. Also, the entities types supported was decided by the knowledge base they used. SemTag used the TAP (Ellen and Jessica 1997) as its knowledge base, which contains a broad range of lexical and taxonomic information about 24 popular objects, including music, movies, authors, sports, autos, health, and so forth.

In its first major run, KnowItAll (Etzioni et al. 2004a) extracted over 50,000 class instances, which consists of five entity classes: City, USState, Country, Actor, and Film. For the class City, the best
result they reported has precision 0.98 at recall 0.76; For the class USState, the best result has precision 1.0 at recall 0.98; For the class Country, the best result has precision 0.97 at recall 0.58; For the class Actor, the best result has precision 0.96 at recall 0.85. For the class Film, however, the performance is not that promising, it has the best precision 0.90 at recall 0.27.

Etzioni et al. (2004b, 2005) propose three extensions to improve KnowItAll’s recall, by using pattern learning for domain-specific extraction rules, sub-class extraction to identify sub-classes, and list extraction to locate lists of class instances, which improves the coverage significantly.

Etzioni et al. (2005) is unique compared with other bootstrapping methods in that it does not require hand-tagged seeds, but instead begins with a domain-independent set of general extraction patterns from which it induces a set of seed instances, and limits its input to a set of predicates of a specific domain.

KnowItAll’s pattern learning is inspired by Hearst (1992), which proposes a set of generic patterns that identify a hyponym relation between two noun phrases.

Nadeau et al. (2006) is inspired by KnowItAll (Etzioni et al. 2005), and extend the named entity types from the three common types (person, location, organization) to more than 50 entity types. In their first stage of retrieving the web page, the human input is reduced to a seed of k manually generated entities. The k is empirically set to 4 as suggested by KnowItAll (Etzioni et al. 2005). In the following stage of apply web page wrapper, a supervised learning algorithm is used to classify unknown entities in the current Web page. No manual annotation is required as the algorithm is trained based on the retrieved web page and the positive sample and negative samples are generated from that web page automatically. They describe their system as unsupervised from this point of view. An evaluation with car brands gives results of precision: 0.88, recall: 0.85, and F1: 0.86.

Ritter et al. (2011) addresses the poor performance when applying common NLP tools on Tweets for named entity recognition. To handle the problem of many infrequent entity types, which is unique on Tweets comparing with common news text, they exploit a large list of entities and their types from Freebase as a source of distant supervision. This mechanism allows the use of large amounts of unlabeled data in learning.

Ivanitskiy et al. (2016) uses distributed word representations trained with Word2vec on the Russian segment of parallel corpora used for statistical machine translation to recognize and classify location, person, and organization. Their reported results are better than state-of-the-art for all these three entity types. The human input is limited to unlabeled corpora.

Mosallam et al. (2014) proposes a fully automatic pipeline, which does not involve machine learning techniques, so that no training data is required. The only input is the unlabeled corpus and a knowledge base. The knowledge base, Aleda, of course, requires human expertise to construct. The pipeline begins with text pre-processing, then possible entity mentions in the corpus are spotted by using knowledge base look-up. Next, a disambiguation step is carried out to disambiguate entities and identify unmatched entities by calculating three context words statistics information around the uniquely matched entities. Their strategy is a simple vote generated by combining the outcome of the three statistics if at least two of the three approaches agree on the same decision. They focus on three common entity types, namely, person, location, and organization. Their reported results slightly outperformed Stanford’s NER Classifier (when trained on a small portion of manually annotated data) and Aleda’s dictionary lookup.

Ling and Weld (2012) reduces the work of collecting training data for their fine-grained tagging set of 112 tags (much more than the common entity classes, e.g., person, organization, location or miscellaneous) by operating in an unsupervised manner, which otherwise would be too large to rely on traditional, manual labeling. Specifically, they exploit the anchor links in Wikipedia text to label entity segments with appropriate tags automatically. Next, they use this heuristically labeled training data to train a Conditional Random Field (CRF) model for segmentation. Then a final step is
assigning tags to the segmented mentions, for which they formulate the tagging problem as multi-class, multi-label classification, and use a perceptron algorithm. In their empirical evaluation, they report a 93% boost in F1 score compared to using the Stanford NER tags alone.

Dalvi et al. (2012) describes an unsupervised information extraction technique, named WebSets, that can extract concept-instance pairs from HTML tables in a given corpus. The novelty is it relies solely on the copra itself, the HTML tables, to detect coordinate terms, without human prepared training data. An unsupervised clustering procedure on the table columns that contain overlapping triplets of the instance will merge table columns and yield sets of entities, each of which potentially belongs to a coherent concept. Since it is a clustering-based algorithm, their experiments are evaluated using three criteria. (1) Are the clusters generated by WebSets meaningful? The results indicated that 63–73% of the clusters were labeled as meaningful. (2) How good are the recommended hypernyms? The results of this test can reach 50–60% of correctness. (3) What is the precision of the meaningful clusters, the coherency of the clusters? The results at this test were in the range of 97–99%.

Swain and Cole (2016) presents a pipeline including tokenization, part-of-speech tagging, named entity recognition, and phrase parsing for chemical information extraction. The named entity recognition phase is finished in an unsupervised manner by using word clustering based on a massive corpus of chemistry articles, and they report improved performance, affording a F-score of 93.4%, 86.8%, and 91.5% for extracting chemical identifiers, spectroscopic attributes, and chemical property attributes, respectively.

Some quick observations include the following:

— Named entity recognition task can be successfully finished in a fully automatic or unsupervised manner. Some of the early systems can already reduce the human efforts required to a very limited level. These early systems were achieving full automation mainly by utilizing a bootstrapping-based approach starting with small initial inputs, like an initial list of the website (Dingli et al. 2003b), a small set of seed examples (Etzioni et al. 2005; Nadeau et al. 2006). Recent systems mainly adopting unsupervised machine learning approaches, can further eliminate human input to unlabeled training data (Dalvi et al. 2012; Mosallam et al. 2014), or even generating training data automatically (Nadeau et al. 2006).

— Some early works, like Armadillo (Dingli et al. 2003b), extracts only small types of general entities (People, Location, Organization). Following works extend the entity types to cover more general objects, like SemTag (Dill et al. 2003b) covers 24 popular objects, Nadeau et al. (2006) extends to more than 50 entity types and even 112 fine defined tags in Ling and Weld (2012). However, regarding the richness of real entity types, the research toward open entity recognition, which can learn and discover new entity types, is strongly required.

Dalvi et al. (2012) is an effort in this direction. It can discover sets of entities from HTML tables in a given corpus, without human prepared training data. The recently popular deep neural network can also be utilized for unsupervised discovery of entities not predefined. Gupta and Manning (2015) uses word embedding, generated from a neural network model Skip-Gram (Mikolov et al. 2013b), to expand entity sets, which can be used for the bootstrapped pattern-based entity extraction system.

A detailed examination regarding named entity recognition using deep learning models can be found in Yadav and Bethard (2018).

— Many works go further into some specific domains, such as Sci-Tech compound entity recognition (Yan et al. 2016), biomedical named entity recognition (Song et al. 2018), entity extraction from clinical records (Alicante et al. 2016; Boytcheva 2018; Henriksson et al. 2015), chemical named entity recognition (Swain and Cole 2016; Zhang et al. 2016),
educational term extraction (Conde et al. 2016), cybersecurity concepts extraction (Xiao 2017), Drug Name Recognition (Liu et al. 2015b), medical named entity recognition (He and Kayaalp 2008; Kavuluru et al. 2013; Skeppstedt 2014), and so forth.

— In addition to English, unsupervised Named Entity Recognition has been studied in other languages, including Chinese (Jia et al. 2018), Spanish (Copara et al. 2016), French (Mosallam et al. 2014), Italian (Alicante et al. 2016), and Russian (Ivanitskiy et al. 2016), as well as cross-lingual (Abdel Hady et al. 2014), even including a dead language like Sumerian (Liu et al. 2015a).

4.1.3 Relation Extraction. After entities are recognized, the Relation Extraction sub-task continues to find and determine the relationship between those entities.

Four semantic annotation tools doing relation extraction task in a fully automatic or unsupervised manner, namely, DIPRE, Snowball, AeroDAML, PANKOW, are selected from Table A.5 to compare with works identified in the literature set. Like in the previous subsection, we list in Table A.7 some notable works regarding the aspects of the human input required, and the performance reported by the authors. For the same reason, we use author-reported results. The selection criteria are the same as that in the previous subsection.

Hearst (1992) was an early work applying linguistic patterns to automatically discover hyponymy lexical relationships between noun phrases. Though it can operate automatically, it requires much human expertise to design the patterns and the lexical relations discovered are mainly limited in well-known styles.

DIPRE (Brin 1999) extracts (author, title) relation from the Web. The human input is limited to five seed tuples of (author, title) pairs and the output is a list of over 15,000 books. They use a simple metric to measure the quality of their results. They randomly pick 20 books out of the result list and attempted to verify that they were actual books by manual searching and verification. The reported result is that 19 of the 20 were all real books.

Snowball (Agichtein and Gravano 2000) is directly inspired by the idea of DIPRE, with a similar objective of extracting an (organization, location) relation. The human input is also limited to five seed tuples, and in addition, a general regular expression. They evaluate the experiment using the precision metric by manually examining a sample of their output. They randomly selected 100 tuples from each of the extracted tables and checked whether each of these tuples was a valid organization-location pair or not. Their optimal settings have 93 tuples that are valid, and only 7 are invalid.

PANKOW (Cimiano et al. 2004) extracts instance-concept relation in web pages with the appropriate concept from an ontology. It first scans web pages for phrases that might be categorized as instances of ontology concepts and outputs a set of candidate proper nouns. Then it exploits patterns to form hypothesis phrases. Some patterns are reused from Hearst (1992). It requires a web page and ontology as initial input. The best Accuracy their system obtains is of 24.9%, which is not very promising.

Textrunner (Banko et al. 2007) is an open information extraction system, which means it is not limited to discovering predefined relations. It requires no input from a human. It experiments over a 9 million web page corpus and compares with KnowItAll running on this dataset. It finds 7.8 million facts having both a well-formed relation and arguments. Of those facts, 80.4% were deemed to be correct according to human reviewers. When comparing with KnowItAll, Textrunner’s average error rate is 33% lower than that of KnowItAll, but it finds an almost identical number of correct extractions.

StatSnowball (Zhu et al. 2009) is another open information extraction system from the Web. It is tested on two datasets, namely, Sent500, which contains 500 sentences and each sentence has
one pair of entities (noun phrases), and Web1M, with 1 million records. Their experiment obtains fairly good results on both Sent500 and Web1M.

Gonzalez and Turmo (2009) proposes an unsupervised approach to learning for Relation Detection, based on the use of massive clustering ensembles. They regard the results obtained as excellent given their system gives a 4 point increase in F1 comparing with the upper bound of a then state-of-the-art approach.

Chen et al. (2017) utilizes global vectors (GloVe) (Pennington et al. 2014), one kind of distributional word representations, on a new finding of unsupervised relation extraction, which they refer to as distributional relation representation. Distributional relation representation aims to automatically learn entity vectors and further estimate semantic similarities between these entities. Their method is not limited to extracting the predefined types; They evaluate by comparing with other common distributional representations and achieve the highest average accuracy which is closest to the supervised method. Another example is Bollegala et al. (2015), which studies the problem of learning word representations that capture the semantic relations between entities.

Pandey et al. (2017) also utilizes the distributional word representation. They feed two types of vectors, generated from unlabeled medical corpora by using Word2Vec (Mikolov et al. 2013b) and GloVe models (Pennington et al. 2014), respectively, to a bi-directional long short-term memory (Bi-LSTM) neural network. Their experiment shows that this combination outperforms baseline Bi-LSTM.

Feldman and Rosenfeld (2006) presents a Web relation extraction system, URES, that learns extraction patterns from unlabeled text. Its input is limited to a short description of the target relations and their attributes. An example is Acquisition (ProperNP, ProperNP) ordered, where the word ordered indicates that Acquisition is not a symmetric relation and the order of its arguments matters. Their system performance can double or even triple the recall achieved by KnowItAll (Etzioni et al. 2005) for relatively rare relation instances while maintaining the same level of precision (Rosenfeld and Feldman 2006).

Some quick observations include the following:

— Relation extraction task can be successfully finished in a fully automatic or unsupervised manner. Some of the early systems can already reduce the human efforts required to a very limited level. These early systems achieve full automation mainly by utilizing a bootstrapping approach and redundant information from the Web (Agichtein and Gravano 2000; Brin 1999).

— Early works usually detect limited types of relation, like DIPRE and Snowball targeting (author, title) relation and (organization, location) relation, respectively. Recent open relation extraction systems, like Textrunner (Banko et al. 2007), StatSnowball (Zhu et al. 2009), and REVERB (Fader et al. 2011), are not limited by predefined relations. The adoption of a distributional semantic representation, like Chen et al. (2017), especially the integration with deep neural network (Pandey et al. 2017), will bring in help for open relation extraction, due to the rich semantic information encoded in the word representation.

— Recently, with the popularity of word embeddings in capturing syntactic and semantic relations among words, there are some works that tried to integrate the word embeddings to achieve the semantic boosting. Batista et al. (2015) uses a similar architecture as Snowball, instead replacing the TF-IDF representations with word embeddings to find similar relationships.

4.1.4 Entity Linking. It is widely known that it is difficult to get a universal and comprehensive evaluation regarding entity linking systems since these systems all differ along multiple dimensions and are rarely compared using the same datasets (Ling et al. 2015). A single entity
linking system typically performs very differently for different datasets and domains (Shen et al. 2015).

Hence, in this subsection, we again first present some author-reported comparisons found in the literature that are helpful to get an insight into the state-of-art of the unsupervised approaches. Then like in previous subsections, we list several seminal named entity linking systems regarding the human input required and the performance reported by the authors in Table A.8. For the same reason, we use the author-reported results. The selection criteria are the same as that in the previous subsection.

Ling et al. (2015) presents an unsupervised system, called Vinculum, and compares with two entity linking systems trained on labeled data, AIDA (Hoffart et al. 2011) and WIKIFIER (Cheng and Roth 2013). Their comparison was carried out on nine datasets. The conclusion drawn from their comparison is that their unsupervised entity linking system, Vinculum, is quite comparable with the highest performance achieved by WIKIFIER, (only 0.6% lower), in the overall performance. Looking at the performance per dataset, VINCULUM and WIKIFIER each is superior in four out of nine datasets, while AIDA only tops the performance on one dataset.

Kamalloo and Rafiei (2018) studies the problem of toponym resolution, which assigns location mention in a document to a geographic referent (i.e., latitude/longitude). Their unsupervised system, Context-Hierarchy Fusion, requires no additional information other than a gazetteer and no training data. Their evaluations show that their method outperforms another unsupervised method, TopoCluster (DeLozier et al. 2015), in terms of precision. However, their method falls below when comparing with supervised techniques as expected. Nonetheless, they have shown that the supervised classifier highly relies on the training data and is less capable of generalizing than an unsupervised method. They also demonstrate that their system outperforms two commercial products, Reuters OpenCalais and Google Cloud Natural Language API, on all three corpora they used.

Kozareva and Ravi (2011) studies the problem of resolving ambiguity associated with names found on the Web. An unsupervised topic modeling framework is adopted to capture hidden information of the entities (senses), which they believe is helpful for the disambiguation. Their experiments demonstrate substantial improvements of 15–35% in task accuracy over an earlier work (Pedersen et al. 2006) adopting the unsupervised clustering approach.

Chen and Martin (2007) also explores the problem of personal named-entity disambiguation. Their method combines robust syntactic and semantic features into unsupervised clustering of documents that contain names. They compare with an earlier model (Bagga and Baldwin 1998) on the corpus also from (Bagga and Baldwin 1998). They find that the use of robust syntactic and semantic features can significantly improve the disambiguation performance for personal names for both Chinese and English.

Mann and Yarowsky (2003) is another work exploring the problem of distinguishing the real world referent of a given name in context. Their approach utilizes an unsupervised clustering technique over a rich feature space of biographic facts. Performance is evaluated based on both a test set of hand-labeled multi-referent personal names and via automatically generated pseudonames.

Zhang et al. (2018) explores the entity resolution problem on three datasets from different domains: Restaurant, Product, and Paper. They propose an enhanced content-based similarity measure to reflect their observation that in some specific domains, terms usually exhibit different discrimination power. Their method works in a fully unsupervised way, consisting of two components, namely, ITER (Iterative Term-Entity Ranking) and CliqueRank. They compare with 14 term similarity measuring algorithms, and their reported results show that their unsupervised framework is comparable or even superior to state-of-the-art methods on the three datasets.

Zhang et al. (2017) builds an online bilingual entity linking system XLink, which is based on Wikipedia and Baidu Baike. They also propose an unsupervised generative probabilistic method.
and utilize text and knowledge joint representations to perform entity disambiguation. They use CoNLL, a popular named entity disambiguation dataset (Hoffart et al. 2011), to test the performance of XLink. Experiments show that the system gets a state-of-the-art performance and a high time efficiency comparing with five other algorithms who carried out their experiments using the same dataset.

Wang et al. (2017b) considers the problem of Named Entity Disambiguation for questions in community question answering (CQA), where there are limited contextual words alongside entity mentions because questions are short. They propose a topic modeling approach which can incorporate prior knowledge learned from the metadata of CQA and Wikipedia into learning and performs learning in an unsupervised manner. They crawled English questions from Yahoo! Answers and Chinese questions from Baidu Knows to establish their datasets. They compare their method with other topic model-based methods, the traditional LDA model (Blei et al. 2003), and another entity-topic model (ET model) (Han and Sun 2012), which also adopts a topic model for entity linking. Their experimental results show better performance.

The ET model proposed in Han and Sun (2012) is an unsupervised model based on a generative model, which can jointly model and exploit the local information (including the words and the mentions in a document) and the global knowledge (including the topic knowledge, the entity context knowledge, and the entity name knowledge). They compare their method with five baselines on two standard datasets and achieve competitive performance.

Dai and Storkey (2011) tackles the problem of referencing authors with real underlying identities. They also describe a generative approach in a completely unsupervised context, where both the abstract of a document and its list of authors are associated with a latent group. They use the author lists and abstracts from several standard publicly available citation databases, such as Citeseer and Rexa, as datasets. Their results show better performance than the best performing unsupervised methods.

Fernández et al. (2012) presents an unsupervised algorithm for entity linking, named WikiIdRank. Their algorithm was implemented and evaluated in the KBP 2010 with positive results. Later on, Jiménez et al. (2013) proposes a set of modifications to WikiIdRank, in order to improve its performance. The different modifications were evaluated in the official KBP 2010 corpus, showing that the best combination increases the accuracy of the initial system in 7.04%. The resultant system, named WikiIdRank++, is unsupervised. A comparison with other approaches in KBP indicates that the system would rank as 4th (out of 16) in the global comparison, outperforming other approaches that use human supervision.

Shen et al. (2012a) presents LINDEN, a novel framework which leverages the semantic information from Wikipedia and the taxonomy of the knowledge base to deal with the entity linking task. They evaluate LINDEN over two public datasets, CZ dataset (Cucerzan 2007) and the TAC-KBP dataset (McNamee and Dang 2009). The experimental results show that LINDEN can achieve significantly higher accuracy on both datasets compared with the state-of-the-art methods.

The DBpedia Spotlight (Mendes et al. 2011) is a system for automatically linking entities in the text with DBpedia URIs. Its automatic pipeline consists of four stages. Among them, each stage can happen automatically without human intervention. We regarded it as fully automatic because we believe it has not employed a specific machine learning algorithm during this course. Their evaluation gives the best results of accuracy 0.85 against other baselines.

Some quick observations include the following:

— Entity linking task also can be finished in a fully automatic (Mendes et al. 2011) or an unsupervised manner (Shen et al. 2012b). The human intervention required can be limited to the minimum level.
Early systems, like DBpedia Spotlight (Mendes et al. 2011), resolves the disambiguation of entities by transforming the DBpedia resource into a vector space and employing standard information retrieval, TF-IDF. Comparing with the TF-IDF technique, which is based on the literal word matching, more advanced machine learning techniques, like topic models, which can solve the polysemy problem due to its excellence inference of latent variable and nature of easy extension, is adopted in recent work (Abeyruwan et al. 2013; Bhattacharya and Getoor 2006; Han and Sun 2012; Liao et al. 2014; Wang et al. 2017b).

4.1.5 Ontology Development. We adopt the method in Maynard et al. (2016) to establish the ontology development into several main tasks: ontology learning, population, and refinement. For these tasks, the most challenging bottleneck is knowledge acquisition from unlimited free text (Abeyruwan et al. 2013; Oliveira et al. 2012; Sánchez 2012). The knowledge can be at different granularity, i.e., class (Tomaz et al. 2012), entity (Ciaramita et al. 2008), concept (Hoxha et al. 2016; Karoui et al. 2007), term (Conde et al. 2016; Maynard et al. 2016), and relation (Punuru and Chen 2012; Sánchez and Moreno 2006). After knowledge acquisition, then follows the slot filling (Fernández et al. 2012), which adds the knowledge from the previous step into an ontology. There are various perspectives to examine the ontology development, like Buitelaar and Cimiano (2008) and Lehmann and Völker (2014). As our concern is the unsupervised approach, we limit our focus on the methods based on NLP, which build on the sub-tasks we have described in previous sections: i.e., named entity recognition, relation extraction, and entity linking. Hence, we do not go into detail in this section.

4.1.6 Summary. By establishing the semantic annotation layer cake, as identified in Figure 4, and by examining three key tasks at different layers, namely, entity recognition, relation extraction, and entity linking, a straightforward conclusion that answers research question 1 is that at each layer and at each sub-task in that layer, we can find unsupervised implementation in literature with fair-to-good performance.

A common practice is doing semantic annotation at both entity and relation layers. There are also efforts to try to achieve a more complex goal, such as expand vertically along the layer cake to provide more comprehensive and generalizable solutions. Along the vertical orientation, the Never-Ending Learning paradigm (Mitchell et al. 2015, 2018) tries to construct a comprehensive pipeline that spans all layers and never stops learning. One of the advantages of Never-Ending Learning is its low demand for pre-processed data required for the learning process.

An implementation of Never-Ending Learning (NEL) (Carlson et al. 2010) learned to read the Web with minimal input at the beginning and rare human intervention during its execution. Its characteristics can be summarized as follows:

- An initial knowledge base defining hundreds of concepts and relations between instances of these concepts.
- A huge unlabeled corpus.
- Occasional interaction with humans during runtime.
- Running 24 × 7.
- Extracting new beliefs from the Web, revising old incorrect beliefs if necessary, and populate the knowledge base.

Another complex practice is the Open Information Extraction (OIE) strategy (Elsahar et al. 2017; Gamallo and García 2015; Jia et al. 2018), which extracts arbitrary open-domain facts of knowledge from massive corpora, instead of being constrained to an existing template stored in an ontology or knowledge base.
4.2 Results Regarding Research Questions 2 and 3

To answer research questions 2 and 3, this section categorizes the methods observed in the literature set and analyzes their properties.

The most obvious observation is that in the literature set generated using the systematic literature method described earlier, the term “unsupervised” has a much higher occurrence than the term “fully automatic.” This indicates that machine learning techniques dominate mainstream efforts to achieve automation for semantic annotation.

The general lifecycle for unsupervised semantic annotation that can be concluded from the literature consists of four main components, i.e., Data, Feature, Model, and Ontology, as shown in Figure 5.

The Data component is responsible for data preparation, pre-processing, and data acquiring. The data acquiring ability is vital to drive the semantic annotation cycle run continuously. The human intervention at this component should be limited to providing an unlabeled corpus.

In the Feature component, features are generated from the data. These features are used as heuristics to guide the unsupervised learning models in the Model component. The heuristics could be as follows:

— Consistencies that exist in different aspects, for example, linguistic (Cimiano et al. 2004; Jia et al. 2018; Sánchez et al. 2011), syntactic, or structural (Krapivin et al. 2008; Sleiman and Corchuelo 2014).
— Information redundancy that exists in various sources (Agichtein and Gravano 2000; Brin 1999; Dill et al. 2003b; Etzioni et al. 2005).
— Latent semantic similarity like co-occurrence (Rosenfeld and Feldman 2007; Rozenfeld and Feldman 2006) or distributional representation (Gupta et al. 2018; Remus 2014; Skeppstedt 2014; Zhang et al. 2016).

Among them, the distributional representation, which is based on the distributional hypothesis, is a newly widely adopted technique that can encode syntactic and semantic info among words in an unsupervised manner.

In the Model component, the features are fed to a concrete model to finish a semantic annotation task. These models can be categorized into different approaches, which include, but are not limited to, bootstrapping, frequency statistics, clustering, graph ranking, neural networks, and hybrid approaches.
4.2.1 BootStrapping Learning Approach. The bootstrapping approach has been widely recognized and frequently employed for various information extraction applications. Strictly speaking, bootstrapping does not fall into the scope of machine learning, but it is one of the earliest methods that can successfully reduce the need for human expertise to a minimum level. Besides, its incremental development from initial seeds can be regarded as a special kind of unsupervised approach.

The bootstrapping approach often operates in a loop flow, consists of several common components, those being Seed, Extract, Generate and an optional Knowledge Base (Ontology), which can assist the pattern or rule learning in the Generate module. The loop is summarized in Figure 6. The starting point of the iteration is the seed instance, which is usually provided by humans, then the Extract module extracts entities or relations by scanning the documents to find occurrences. Based on the newly found instances, and optionally may be under the help of the external knowledge base, the Generate module generates new patterns or rules, which are then passed to the Extract module to start a new iteration of the process. The cycle stops once specific criteria are reached such as no more entities or relations being found, specific information being found, or other user requirements are satisfied. In the Never-Ending Learning paradigm (Mitchell et al. 2015, 2018), the loop continues running.

The bootstrapping approach is essentially a self-learning process that tries to learn new instances based on the seed instances. This characteristic means that the bootstrapping approach is most readily applicable for tasks that learn new knowledge that has strong integrity with that provided in the seed examples.

For example, DIPRE (Brin 1999) extracts (author, title) relation from the Internet based on a seed set provided by users.

Similarly, Snowball (Agichtein and Gravano 2000) extracts a structured organization-location relation, with some user-provided example organization-location tuples provided as seeds.

Mukherjee et al. (2005) annotates unlabeled concepts in new HTML pages through the bootstrap algorithm started with a small set of hand-labeled concept instances. Their approach is based on the assumption that an HTML page usually exhibits consistency in presentation style and spatial locality, especially template-based content-rich documents such as the e-commerce sites.

Armadillo (Chapman et al. 2004; Ciravegna et al. 2004; Dingli et al. 2003a) presents an application for finding people’s names from some websites of Computer Science Departments. This name-finding task is triggered by a small set of seed names.

Sahay et al. (2008) also uses the bootstrapping method with a similar workflow to DIPRE and Snowball. The difference is that it only examines the snippets of the search result to reduce the amount of the text that needs to be examined.
The human expertise involved in the bootstrapping approach is usually limited to handcrafted seed instances or rules/patterns. In addition to the manually created training seeds, some methods also require some kind of knowledge base, such as an ontology to help in the learning process. For example, Armadillo (Chapman et al. 2004; Ciravegna et al. 2004; Dingli et al. 2003a) uses an ontology to help identify which items on a web page are instances of concepts in the ontology.

Compared to other bootstrapping methods, KnowItAll (Etzioni et al. 2005, 2008) completely removes the seed examples that other bootstrapping methods require. Instead, it induces a set of seed instances from a general extraction patterns set. Though successfully removing the need for seed examples, KnowItAll still relies on a laborious bootstrapping process for each relation of interest.

A system closely related to KnowItAll is URES (Feldman and Rosenfeld 2006; Rosenfeld and Feldman 2006). KnowItAll is good at finding the relations that are frequently mentioned in a wide variety of sentences. In contrast, URES is more effective in detecting relations that are relatively infrequent.

Several works are claiming to be wholly unsupervised but adopting the bootstrapping method. For example, Cui et al. (2010) does not need training examples or annotation templates, with the manual labor required limited to compiling dictionaries and preparing source documents.

Munro and Manning (2012) also do not need manually labeled items, or external resources. The only resource is parallel text which provides cross-language edit likelihood to bootstraps to make broad predictions across both languages.

Uszkoreit et al. (2010) provides an analysis of the effects of certain properties of data and the selection of seeds toward the performance of some general relation learning framework.

Fujiwara and Sekine (2011) presents a self-learning bootstrapping, where the seed set is divided into the training set and validation set. The bootstrap initially starts with the training set, trying alternative parameter settings, and uses the validation set to identify the optimal settings. This is done iteratively with different segmentations of the original set using cross-validation.

As Wu et al. (2017) points out, most bootstrapping techniques are based on syntactic patterns, i.e., each iteration finds more syntactic patterns for subsequent extraction. Another technique is the semantic bootstrapping.

Recently, with the popularity of word embeddings in capturing syntactic and semantic relations among words, there are some works that tried to integrate the word embeddings to achieve the semantic boosting. Batista et al. (2015) uses a similar architecture as Snowball, instead replacing the TF-IDF representations with word embeddings to find similar relationships. Wu et al. (2017) presents a detailed analysis of semantic bootstrapping from a theoretical perspective to better understand the working mechanism of semantic bootstrapping, which otherwise remains elusive.

Gupta and Manning (2015) uses word vectors to expand entity sets, which can be used for the bootstrapped pattern-based entity extraction system.

4.2.2 Frequency Statistics-Based Approach. This approach usually utilizes the redundancy of information to validate a prediction. This is a very natural choice when similar or duplicate information is available from multiple sources, like in the World Wide Web. This method was adopted early on and is still used.

Both PANKOW (Cimiano et al. 2004) and KNOWITALL (Etzioni et al. 2005) were early endeavors utilizing frequency statistics to accomplish unsupervised information extraction from the Web.

PANKOW (Cimiano et al. 2004) discovers entities such as cities, countries, and so forth, via the Google API. It scans web pages for phrases that might be categorized as instances of ontology concepts and outputs a set of candidate proper nouns. It then builds hypothesis phrases by iterating through all the candidate proper nouns and candidate ontology concepts. After that, it queries...
Google with the hypothesis phrases and counts the hits for each one. The candidate proper nouns are then categorized into the highest ranked concepts as the annotation for that piece of text.

KNOWITALL (Etzioni et al. 2005) generates candidate facts based on a pattern set, which contains eight domain-independent extraction patterns, then it uses the frequency statistics returned from search engine queries to determine the most likely member of the class.

Compared with PANKOW, Alchymoogle (Sætre et al. 2005) works in the more specific domain of biomedicine, which uses 275 different semantic tags, much more than the 59 concepts for general phrases in the ontology used by PANKOW. Similarly, Alchymoogle utilizes Google for semantic annotation (extracting is-a relations specifically).

Like Alchymoogle, Sahay et al. (2008) also discovers biomedical relation from the Web but can also identify various relationship types by utilizing a pattern database. It also uses the WordNet and UMLS ontologies to determine synonyms for given resources to increase coverage. In addition to the general Web search engines, it also utilizes specific domain databases like PubMed.

Due to the openness of the World Wide Web, information in vast amounts may not necessarily equate to high quality. A solution is to combine information from other, more trustworthy sources. Yan et al. (2009) is an example, combining two sources, Wikipedia articles and a Web corpus, for a relation extraction task. Their result demonstrates the superiority of complementary pattern combination over single source approaches.

McDowell and Cafarella (2006) introduces OntoSyphon, which operates in an ontology-driven manner: taking any ontology as input, OntoSyphon uses the ontology to specify web searches that identify possible semantic instances, relations, and taxonomic information. Redundancy in the Web, together with information from the ontology, is then used to automatically verify these candidate instances and relations, enabling OntoSyphon to avoid unnecessary noise from mixed Web content.

### 4.2.3 Clustering-Based Approach

Clustering is a premium unsupervised algorithm in the machine learning community. The principal rationale behind the clustering algorithm is to achieve a state such that samples within the same cluster are logically/semantically close to each other and are different from samples in other clusters. As a kind of unsupervised algorithm which runs in a context without labeled data, a heuristic is needed to guide the partitioning and formation of the clusters. The heuristics observed in the literature set are mainly based on some kinds of similarity, including the following:

- The similarity of the contexts around entity pairs (Chen et al. 2005).
- The similarity of internal hypertext structure (Salin et al. 2015), structural and visual features of web page elements (Walther 2012).
- The duality between the extensional and intentional representation of semantic relations (Bollegala et al. 2010).
- Co-occurrence of entity pairs based on the contexts in which they appear (Rosenfeld and Feldman 2007; Rozenfeld and Feldman 2006).
- Overlapping between triplets of instances (Dalvi et al. 2012).
- The similarity between parse trees (Zhang et al. 2005).
- Semantic similarity (Wang et al. 2011).

Some other heuristics include the integrated signals from multiple information sources in Li et al. (2013a)’s entity attribute synonyms task and features generated by dependency parsing on a sentence containing entities (Akbik et al. 2012).

Besides the premium clustering algorithm, \(k\)-means (Akbik et al. 2012; Verbanck et al. 2013), other clustering algorithms observed include the self-organizing map (SOM) (Kohonen et al. 2000;
Yang (2006, 2009), and several hierarchical clustering algorithm, i.e., hierarchical SOM (Chifu and Letia 2010), LIMBO (Andritsos et al. 2004), and Brown clustering (Swain and Cole 2016).

4.2.4 Graph-Based Approach. Graph-based algorithms are another important component of the machine learning community. In the literature set, we observed many works adopting a graph-based approach.

Hassan et al. (2006) presents an unsupervised approach based on graph mutual reinforcement for information extraction. Their approach does not require any seed patterns or examples. Instead, it depends on redundancy in large datasets and graph-based mutual reinforcement to induce generalized "extraction patterns."

Duque et al. (2018) presents a graph-based unsupervised technique to address word sense disambiguation in the biomedical domain. The graph is built with co-occurrence information from medical concepts found in scientific abstracts, and disambiguation is performed using the personalized PageRank algorithm.

Salin et al. (2015) presents a framework for unsupervised domain- and language-independent annotation of the web page. They represent the hypertext structure as a graph and apply clustering algorithms to the graph to produce a set of web-page clusters. These clusters can be further used for annotating the web pages.

Bougouin et al. (2016) reduces the semantic annotation to key-phrase annotation. Key-phrase annotation is carried out by extracting the most important phrases from a document as its annotation. They use two graph representations, one for the document and one for the controlled vocabulary from the specialized domain. Then apply a co-ranking algorithm to perform both key-phrase extraction and assignment in a mutually reinforcing manner.

Vidal et al. (2014) builds graphs differently. Each relevant term of the document is related with not only a concept in the ontology but also an instance which is expanded to a predefined depth limit. These (sub)graphs provide a better description of the document.

4.2.5 Neural Network-Based Approach. Recent works show that Neural Network models, especially the deep neural network, obtain state-of-the-art performance in the several sub-tasks. One attractive property of the deep neural network is that it can automatically learn features from text corpora without complicated pre-processing and minimize the dependence on manual expertise.

We first introduce several works based on deep neural networks for the several sub-tasks established earlier. Then we specifically introduce the word embedding and its successful application in various sub-tasks.

Wang et al. (2017a) develops an Algorithm of Formal Ontology Generation (AFOG) via deep machine learning to enable autonomous generation of ontologies.


Among the many neural network models, we observed a shallow neural network with straightforward architecture, Word2vec (Mikolov et al. 2013b), is widely adopted in various tasks. Though many works regard Word2vec as a deep neural network (Albukhitan et al. 2017; Casteleiro et al. 2018; Kolyvakis et al. 2018), it strictly is not deep. It consists of only three standard layers, i.e., input layer, hidden layer, and output layer. Its hidden layer has no activation function, instead only projection. This is different from deep learning, which usually has multiple non-linear middle layers.

Word2vec is particularly computationally efficient for learning word embeddings from raw text. The word embedding is based on the distributional hypothesis, “You shall know a word by the company it keeps” (Firth 1957). Word2vec comes in two flavors, the Continuous Bag-of-Words model.
The CBOW model serves the assumption that a context ought to be able to predict its missing word. Meanwhile, the Skip-Gram model serves the assumption that a word ought to be able to predict its context.

The word embedding generated by Word2vec can capture some general, and in fact quite useful, semantic information about words and their relationships to one another, like (Capital, Country) relationships, (Paris, France), (Rome, Italy), and so forth; verb tense, (walking, walked), (swimming, swam) (Mikolov et al. 2013b). The most famous and interesting relation captured is the equation “King − Man + Woman” results in a vector very close to “Queen” (Mikolov et al. 2013c). These relationships can be easily observed using a simple arithmetic of the embedding vectors, i.e., vec(“man”) − vec(“woman”) + vec(“king”) is closer to vec(“queen”) than to any other word vectors. Similarly, vec(“Madrid”) − vec(“Spain”) + vec(“France”) is closer to vec(“Paris”) than to any other word vectors (Mikolov et al. 2013a, 2013b). The benefits of Word2vec include (1) the relations acquired are not limited to pre-defined categories and (2) the word embedding representations are trained directly from the raw natural text without human supervision; not even complex pre-processing, is fully unsupervised.

We choose some works utilizing Word2vec in the semantic annotation sub-tasks:

— Xiao (2017) first locates a named entity by using any existing NER system, then acquires word vectors by applying unsupervised model Word2vec extraction upon domain texts and ontologies, and then feeding these features into a voting-based model to classify the marked named entities into fine-grained classes.

— Sikdar and Gambäck (2017) also builds word vectors using the unsupervised learning algorithm, Word2vec, then trains a CRF classifier to predict each token’s named entity class.

— Pandey et al. (2017) initializing the training of their long short-term memory (Bi-LSTM) neural network with pre-defined clinical word embeddings using Word2Vec and GloVe from widely available medical resources.

— Ivanitskiy et al. (2016) uses distributed word representations trained with Word2vec on the Russian named entity recognition and classification.

— Kolyvakis et al. (2018) presents a representation learning approach that is tailored to the ontology alignment task. Their approach is based on embedding ontological terms in a word vector to capture semantic similarity.

— Albukhitan et al. (2017) uses the CBOW and Skip-gram models to perform the Arabic ontology learning tasks.

— Casteleiro et al. (2018) investigates the use of CBOW and Skip-gram on the task of identifying term variants or acceptable alternative free-text terms for gene and protein names from the millions of biomedical publications, with the support of knowledge formalized in Cardiovascular Disease Ontology (CVDO). Their study shows performance improvements for both CBOW and Skip-gram on this gene/protein synonym detection task.

The word embedding generated by the Word2Vec model can also be used as a pre-processing step, after which the learned vectors are fed into a downstream model (typically a deep neural network) to generate predictions and perform all sorts of exciting things. Sikdar and Gambäck (2017) feeds the word vectors generated by Word2vec to a deep neural network to assign labels to the words.

Another example of distributional word representation which also obtains encouraging performance is the Glove (Pennington et al. 2014).

It is notable that although unsupervised approaches may tend to perform weaker than supervised approaches in terms of precision, they usually have a higher recall like the neural networks
4.2.6 Probabilistic Model Approach. Probabilistic topic generative models are a field of probabilistic modeling, in which data are generated from a process that includes hidden variables. The generative process defines a joint probability distribution over both the observed and hidden random variables. The data analysis uses joint distribution to compute the conditional distribution of the hidden variable given the observed variables.

With the wide adoption of probabilistic topic modeling to discover thematic structure, and further use of annotation to visualize, organize, and summarize documents, semantic annotation is often carried out at the topic-labeling level, which aims to assign meaningful labels for discovered topics.

Among the probabilistic topic models, the most adopted one is the Latent Dirichlet Allocation model (Blei et al. 2003).

Chien and Cheng (2015) utilizes a probabilistic model for the task of semantic tagging of mathematical expressions. The user habits of writing expressions is regarded as a hidden variable and is considered with the variable properties in an integrated framework. Their method requires no labeled training data but only our mathematical grammar set.

Wang et al. (2018) designs two generative probabilistic models, which exploit more distinguishable n-gram features beyond the limitation of Rel-LDA and Type-LDA to extract a relation from unstructured biomedical text.

Chemudugunta et al. (2008) presents a probabilistic modeling framework that combines both predefined ontologies and data-driven topics in a principled manner. It automatically tags web pages with concepts from a pre-defined concept set without any need for annotated documents.

Konkol et al. (2015) proposes new features for Named Entity Recognition based on latent semantics. Their features are learned by combining local and global context, in which the global semantic information is learned via topic models.

Liu et al. (2013) employs an unsupervised method using topic models to discover the distribution of features of latent relations. They compute the distinctiveness of features by using the obtained feature-relation distribution and assign feature weights based on their distinctiveness to train the extractor.

Yates and Etzioni (2009) presents a system called Resolver, which introduces a probabilistic relational model for the task of identifying synonymous relations and objects, or synonym resolution, predicting whether two strings are co-referential based on the similarity of the assertions containing them.

4.2.7 Hybrid Approach. We have presented several categories of unsupervised approaches for semantic annotation. Each approach has its characteristics and can work well in their particular optimal setting. However, many of these approaches can be combined for better performance. This subsection investigates research works that try to do precisely this.

The iterative increasing mechanism of the bootstrapping approach fits well with the loop of the lifecycle in Figure 5 and can integrate with the other approaches to achieve the long-term learning. Souza et al. (2008) is a combination of the bootstrapping approach and clustering approach. It starts from a small number of user-defined seeds and bootstraps itself by exploiting a novel clustering technique. The clustering algorithm they used is LIMBO (Andritsos et al. 2004).

Another widely adopted choice in combination is the frequency statistics approach, which can provide useful heuristics, such as the redundancy information, to guide unsupervised learning.

These two approaches, bootstrapping, and the frequency statistics can be easily integrated to increase both automation and performance. Examples of this combination include KNOWITALL.
(Etzioni et al. 2005), Armadillo (Dingli et al. 2003b), PANKOW (Cimiano et al. 2005), SemTag (Dill et al. 2003b), and (Sahay et al. 2008).

Another typical combination with the frequency statistics approach is the clustering approach. For example, Mohamed et al. (2011) utilizes the redundancy of information in Web data to cluster context patterns which are semantically similar. Another example is Salin et al. (2015), which represents the hypertext structure of a web page as a graph and applies clustering algorithms on the graph to produce a set of web-page clusters based on the redundancy among similar web pages.

Some works integrate more than two approaches. For example, Hassan et al. (2006) combines three approaches: frequency statistics approach, graph-based approach, and clustering approach. It constructs generalized extraction patterns based on the redundancies in a huge body of text data. It then identifies the most informative patterns by deploying on a mutual reinforcement bipartite graph consisting of patterns and tuples. A clustering algorithm, Markov Cluster Algorithm (MCL), is applied to the graph to eliminate weak edges and produce clusters for similar tuples.

4.2.8 Summary. We identified several groups of unsupervised approaches observed in the literature. They are BootStrapping Learning approach, Frequency Statistics–Based approach, Clustering-based approach, Graph-based approach, Neural Network–based approach, Probabilistic Model–based approach, and Hybrid approach. The bootstrapping approach, strictly speaking, is not a machine learning algorithm, and is indeed a paradigm that provides mechanical support for continuous learning. The frequency statistics–based approach is the approach for feature generation from data, which can be used to guide the unsupervised algorithms.

5 CONCLUSION AND FUTURE DIRECTION

By carrying out a new literature review, we find that there has been insufficient work regarding the surveying of semantic annotation approaches from the perspective of the degree of automation, although numerous works do exist targeting the objective of reducing human/expert intervention in the semantic annotation procedure. In this study, we mainly focus on the top degree of automation.

By examining the literature, we also find an inconsistent mess of terminology used in the literature to describe the various semantic annotation tools’ degree of automation. We propose consistent terminology to describe the degree of automation, considering in particular whether a machine learning technique is adopted or not. Based on this scale, we uniformly ascertain the degree of automation of the many semantic annotation tools that were already investigated in previous survey papers.

To understand the state-of-the-art of unsupervised approaches for doing semantic annotation tasks, this work addressed three research questions:

Question 1. To what extent can semantic annotation be completed in a fully automatic or unsupervised way?

Answer 1. Motivated by the first research question, this study identified a taxonomy of sub-tasks or problems within the scope of semantic annotation. Though this taxonomy may not be entirely comprehensive and may be subject to change as new research directions appear, it provides the insight necessary to understand the procedure of semantic annotation better. By analyzing several main individual sub-tasks (named entity recognition, relation extraction, entity linking, and ontology development), and examining the complete workflow of annotation (i.e., the Never Ending Learning paradigm), we understand and try to justify that semantic annotation can not only achieve a high degree of automation at some specific tasks but can also achieve a completely automated workflow of annotation at a limited scale.
Question 2. What kind of fully automatic or unsupervised approaches for semantic annotation exist in literature?

Answer 2. This work summarizes and categorizes the methods used in the surveyed literature set into seven main groups, namely, BootStrapping-based approach, Frequency Statistics–based approach, Clustering-based approach, Graph-based approach, Neural Network–based approach, Probabilistic Model approach, and Hybrid approach.

Question 3. What characteristics and relation do these approaches have?

Answer 3. Throughout this article, we surveyed a wide range of approaches, of several different classes and in some cases hybridizing those classes, in order to provide a representative sample of the present state-of-the-art. We also analyze the characteristics of the approaches in different categorizations.

Trends/Open Questions. Although semantic annotation has been a hot research topic for decades, fully automated semantic annotation, though anticipated as a solution to overcome the large volume of data requiring annotation, still has a long way to go before achieving the results necessary to play its role in the vision of Semantic Web, and many important research questions remain open:

—As observed through this study, full automation can be achieved locally in the context of some single sub-tasks for some specific application with related coverage. Higher-degree automation that chains more than one sub-task or the whole procedure together is expected in future research work. Mitchell et al. (2015, 2018) is a representative example that provides a comprehensive solution covering all sub-tasks under the semantic annotation scope. Based on it, more robust solutions and more accurate results can be expected to arise in future research.

—Based on the observation in previous sections, especially Sections 4.1.2, 4.1.3, and 4.2.5, which summarize the sub-tasks of Named Entity Recognition, Relation Extraction, and the Neural Network–based approach, respectively, we can say that the broad adoption of the neural network approach, especially the Word2vec model, which can be seen as a simplified deep neural work with only three layers, brings promising results in various sub-tasks by learning distributional representation of text snippets (word, entity pairs, context, etc.).

—We want to say neural network approaches, especially the recent deep neural network, has found success in not only flat semantics such as entity or relation, but also go further linked with ontology resources, like knowledge representation, learning, reasoning, as demonstrated in previous sections of entity recognition, relation extraction, entity linking, and ontology development.

—With the recent advances in deep neural networks, we can see more and more forms of representation learning have emerged. This contribution into representational issues promotes the integrative development of neural networks learning and logical reasoning, which further can help to resolve the challenge of learning and reasoning, which is one of the main challenges of Artificial Intelligence (AI) (Valiant 2008). For achievements and challenges along this line of research, readers are referred to Garcez et al. (2015).

—A clear trend that can be observed within this survey is that semantic annotation ranges from basic level with little real semantics involved (occurrence detection, instance detection, etc.) to high-level rich semantic annotation (keyphrases, topics, etc.). Since current ontologies mainly consist of concepts and their relations, the ontology enrichment task should embrace the challenge brought by the knowledge richness from a higher granularity (e.g., the use of keyphrases and topics).
APPENDICES
A SEARCH STRATEGY
A.1 Survey Paper Identification

The following factors are considered when building Query 1:

— Keywords for search are selected according to the following guidelines:
  — The key phrase “semantic annotation” is used since the objective of this search is to find related review papers on this very topic.
  — The keywords “survey,” “review” and some of their synonyms, such as “study” and “examination” are used. “State of the art” is also used since it is a common phrase used in review paper titles.
  — We limit the search to the title, abstract, and assigned keywords of the papers.
  — Scopus provides search fields for “title” and “title-abs-key.”
  — Web of Knowledge provides “Title” and “Topic” search fields. According to the Web of Science Core Collection Help, the “Topic” search field covers the following fields within a record: Title, Abstract, Author Keywords, and Keywords Plus.
  — Google Scholar provides two exclusive options for search fields, either “anywhere in the article” or “in the title of the article.” If the first option is chosen, then the search results will be quite noisy. So when searching Google Scholar, the search is limited to “in the title of the article.”
  — As mentioned earlier, this survey only focuses on semantic annotation in textual media, so we exclude literature with “image,” “video,” “audio,” “visual,” “gene,” “scene,” “imagery,” “vision,” “multimedia,” and “media” (which commonly is not used to refer to simply text) in the title.
  — We also limited ourselves to literature published in English.

The search strings for Query 1 submitted to Scopus, Web of Science, and Google Scholar, respectively, along with the count of results for each search and the final count of documents retained after examination are listed in Table A.1.

<table>
<thead>
<tr>
<th>Digital Libraries</th>
<th>Query 1</th>
<th>Results Count</th>
<th>Final Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopus</td>
<td>(TITLE (&quot;semantic annotation&quot;) AND TITLE (survey OR review OR study OR examination OR &quot;state of the art&quot; OR analyse) AND NOT TITLE (image OR video OR audio OR visual OR gene OR scene OR imagery OR vision OR multimedia OR media OR &quot;case study&quot;)) AND (LIMIT-TO (LANGUAGE, &quot;English&quot;))</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>Web of Science</td>
<td>TITLE: (&quot;semantic annotation&quot;) AND TITLE: (survey OR review OR study OR examination OR &quot;state of the art&quot; OR analyse) NOT TITLE: (image OR video OR audio OR visual OR gene OR scene OR imagery OR vision OR multimedia OR media OR &quot;case study&quot;) Timespan: All years. Indexes: SCI-EXPANDED, SSCI, A&amp;HCI, ESCI.</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Google Scholar</td>
<td>allintitle: &quot;semantic annotation&quot; survey OR review OR study OR examination OR &quot;state of the art&quot; OR analyse -image -video -audio -visual -gene -scene -imagery -vision -multimedia -media -&quot;case study&quot;</td>
<td>33</td>
<td>6</td>
</tr>
</tbody>
</table>

Using “study” as a search keyword will result in literature with “case study,” so we exclude “case study” in the AND NOT condition. If the reader uses the query string in the second row for reproducing results in Web of Science, some syntax change is required, e.g., TITLE should be replaced with TI. Patents and Citations are excluded when searching in Google Scholar.
A.2 Further Literature Search

In addition to the factors considered in building Query 1, the following factors were considered when building the Query 2. The search strings for Query 2 submitted to the three search services, along with the count of results for each search and the final count of documents retained after examination are listed in Table A.2.

— A two-step procedure decided the search keywords. First, a keyword set is generated by aggregating author keywords from each paper of semantic annotation tools listed in the “Semantic Annotation tools” column of Table A.4. Then we selected some additional keywords representing the known sub-tasks or problems in semantic annotation in general. The selected keywords were "semantic annotation," "information extraction," "entity extraction," "entity recognition," "entity linking," "entity disambiguation," "entity resolution," "relation extraction," "relation recognition," "ontology learning," "ontology population," and "ontology refinement."

— As the third observation in Section 3 reveals, the term “automatic" is a very general term which will return many papers only partially automatic. We instead use “unsupervised" and "full(y) automatic" as search keywords.

As a result, a final list of 12 survey papers was aggregated from the Survey Paper Identification phase, which is listed in Table A.3. A final literature set of 367 papers was determined in the Further Literature Search phase.

Table A.2. Query 2 as Submitted to Scopus, Web of Science, and Google Scholar, along with the Count of Results from Each Query and the Final Count of Documents Retained After Examination

<table>
<thead>
<tr>
<th>Digital Libraries</th>
<th>Query 2</th>
<th>Results Count</th>
<th>Final Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopus</td>
<td>(TITLE-ABS-KEY (&quot;semantic annotation&quot; OR &quot;information extraction&quot; OR &quot;semantic tagging&quot; OR &quot;entity recognition&quot; OR &quot;entity extraction&quot; OR &quot;entity linking&quot; OR &quot;entity disambiguation&quot; OR &quot;entity resolution&quot; OR &quot;relation extraction&quot; OR &quot;relation recognition&quot; OR &quot;ontology learning&quot; OR &quot;ontology population&quot; OR &quot;ontology refinement&quot;) AND TITLE-ABS-KEY (unsupervised OR &quot;fully automatic&quot;) AND NOT TITLE-ABS-KEY (image OR video OR audio OR visual OR gene OR scene OR semi OR imagery OR &quot;supervised&quot; OR vision OR multimedia OR media)) AND (LIMIT-TO (LANGUAGE, &quot;English&quot;))</td>
<td>441</td>
<td>198</td>
</tr>
<tr>
<td>Web of Science</td>
<td>TOPIC: (&quot;semantic annotation&quot; OR &quot;information extraction&quot; OR &quot;semantic tagging&quot; OR &quot;relation extraction&quot; OR &quot;relation recognition&quot; OR &quot;entity extraction&quot; OR &quot;entity recognition&quot; OR &quot;entity linking&quot; OR &quot;entity disambiguation&quot; OR &quot;entity resolution&quot; OR &quot;ontology learning&quot; OR &quot;ontology population&quot; OR &quot;ontology refinement&quot;) AND TOPIC: (unsupervised OR &quot;fully automatic&quot;) NOT TOPIC: (image OR video OR audio OR visual OR gene OR scene OR semi OR imagery OR &quot;supervised&quot; OR vision OR multimedia OR media) Timespan: All years. Indexes: SCI-EXPANDED, SSCI, A&amp;HCI, ESCI.</td>
<td>92</td>
<td>30</td>
</tr>
<tr>
<td>Google Scholar</td>
<td>allintitle: unsupervised OR &quot;fully automatic&quot; &quot;semantic annotation&quot; OR &quot;Information Extraction&quot; OR &quot;semantic tagging&quot; OR &quot;entity recognition&quot; OR &quot;entity extraction&quot; OR &quot;entity linking&quot; OR &quot;entity disambiguation&quot; OR &quot;entity resolution&quot; OR &quot;relation extraction&quot; OR &quot;relation recognition&quot; OR &quot;ontology learning&quot; OR &quot;ontology population&quot; OR &quot;ontology refinement&quot; -image -OR -video -OR -audio -OR -visual -OR -gene -OR -scene</td>
<td>173</td>
<td>139</td>
</tr>
</tbody>
</table>
Table A.3. Twelve Survey Papers and their Contribution. Column “Automation” indicates whether that paper investigates the degree of automation of the semantic annotation tools covered in the papers.

<table>
<thead>
<tr>
<th>Survey Papers</th>
<th>Main Contribution</th>
<th>Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP1</td>
<td>Specifically focuses on the annotation of biomedical entity references in biomedical documents and reviewed a selected set of biomedical semantic annotation tools.</td>
<td>✗</td>
</tr>
<tr>
<td>SP2</td>
<td>Summarizes the semi-automatic semantic annotation of Web services. Three research artifacts were addressed: (1) well-defined and adopted ontologies, (2) real case studies and implementations of complex industrial semantic services, (3) tools able to support the annotation of semantic services.</td>
<td>✗</td>
</tr>
<tr>
<td>SP3</td>
<td>Collected and classified semantic annotation survey research. Based on their literature review, they compared the semantic annotation methods inside surveyed research according to the one well-accepted enterprise architecture framework that describes and classifies different knowledge representations in an enterprise, the Zachman Framework, with emphasis on “What,” “Why,” and “How” to perform the semantic annotation.</td>
<td>✓</td>
</tr>
<tr>
<td>SP4</td>
<td>Investigates and groups semantic annotation tools according to various dimensions: (1) storage, (2) the information extraction process, (3) dynamicity, (4) customization, (5) scalability. The information extraction process is further subdivided into (1) the method type, (2) the automation level. Automation level usually falls into three levels, the most basic level using handcrafted rules or wrappers to capture known patterns.</td>
<td>✓</td>
</tr>
<tr>
<td>SP5</td>
<td>Provides a classification scheme with three main dimensions: (1) structural complexity of annotations, (2) type of the vocabularies used, (3) the collaboration type supported. The scheme helps clarify requirements of end-user use cases and simplify the communication between semantic annotation experts and actual users.</td>
<td>✗</td>
</tr>
<tr>
<td>SP6</td>
<td>Examines semantic annotation tools using supervised machine learning methods from the two constituent sub-tasks of semantic annotation: named entity extraction and relation extraction.</td>
<td>✓</td>
</tr>
<tr>
<td>SP7</td>
<td>Defines a framework for examining semantic annotation platform differences based on platform characteristics, surveys several platform implementations, and defines a classification scheme based on the information extraction method used.</td>
<td>✗</td>
</tr>
<tr>
<td>SP8</td>
<td>Identifies seven requirements of semantic annotation tools and assesses the then state-of-the-art capabilities of annotation systems against them. One of the seven requirements is automation. A simple summary of the annotation tools based on the degree of automation is provided.</td>
<td>✓</td>
</tr>
<tr>
<td>SP9</td>
<td>Identifies a common architecture among different ontology-based information extraction systems and classifies them based on different factors, including the information extraction method, ontology construction, update method, components of the ontology extracted, and types of sources.</td>
<td>✗</td>
</tr>
</tbody>
</table>

(Continued)
Table A.3. Continued

<table>
<thead>
<tr>
<th>Survey Papers</th>
<th>Main Contribution</th>
<th>Automation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP11</td>
<td>Aggregates author-reported performance of semantic annotation platforms, including Armadillo, KIM, MnM, MUSE, PANKOW, and SemTag.</td>
<td>✗</td>
</tr>
<tr>
<td>SP12</td>
<td>Provides an overview of available tools utilizing different properties, one of which is the degree of automation.</td>
<td>✓</td>
</tr>
</tbody>
</table>

B COMPARATIVE STUDY OF RELATED WORKS

B.1 Comparison of Surveys

Table A.4. The Semantic Annotation Tools Surveyed by Five of the Twelve Survey Papers Identified in Table A.3 Related to their Degree of Automation

<table>
<thead>
<tr>
<th>Survey Papers</th>
<th>Semantic Annotation Tools</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP6</td>
<td>KNOWITALL (Etzioni et al. 2004a), TEXTRUNNER (Banko et al. 2007)</td>
<td>Unsupervised, Self-supervised learner14</td>
</tr>
</tbody>
</table>

(Continued)

14Since their Learner module labels its own training data, they refer to it as self-supervised.
## B.2 A Uniform Summary of the Degree of Automation

### Table A.5. Uniform Summary of the Degree of Automation of the Notable Annotation Tools Identified in the Survey Papers Listed in Table A.3

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Year</th>
<th>Degree of Automation</th>
<th>User Input</th>
<th>Scenario</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIPRE</td>
<td>1999</td>
<td>Fully automatic</td>
<td>Five seed tuples of (author, title) pairs.</td>
<td>Extract (author, title) relation from the Web.</td>
<td>No</td>
</tr>
<tr>
<td>Snowball</td>
<td>2000</td>
<td>Fully automatic</td>
<td>Five seed tuples of organization-location pairs and a general regular expression.</td>
<td>Extract (organization, location) relation.</td>
<td>No</td>
</tr>
<tr>
<td>AeroDAML</td>
<td>2001</td>
<td>Fully automatic</td>
<td>A user simply enters a URL.</td>
<td>Generate DARPA Agent Markup Language (DAML) annotations for web pages.</td>
<td>Yes</td>
</tr>
<tr>
<td>Cream / Ont-O-Mat</td>
<td>2002</td>
<td>Semi-automatic</td>
<td>Three types of interaction with a user are required.</td>
<td>Creating annotations for web pages using concepts in an ontology.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note, each paper uses its own terminology for the degree of automation, which diminishes clarity and comparability. SP12 (Fähndrich et al. 2015) defined “Semi” as the ability to automatically perform some task with the constraint that there is still the requirement to supervise the process, “Collection” was defined as the ability to automatically collect information since they believe the collection of information is a time-consuming task which needs to be automated.

(Continued)
### Table A.5. Continued

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Year</th>
<th>Degree of Automation</th>
<th>User Input</th>
<th>Scenario</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIM</td>
<td>2003</td>
<td>Semi-automatic</td>
<td>Knowledge Base, Lexicon/Parsing Rules.</td>
<td>Identified Name Entities from a corpus of general news. Ontology aware IE.</td>
<td>Yes</td>
</tr>
<tr>
<td>SemTag</td>
<td>2003</td>
<td>Fully automatic</td>
<td>Knowledge base.</td>
<td>Annotating a collection of approximately 264 million web pages, generating approximately 434 million automatically disambiguated semantic tags.</td>
<td>Yes</td>
</tr>
<tr>
<td>PANKOW</td>
<td>2004</td>
<td>Fully automatic</td>
<td>A web page and ontology.</td>
<td>Annotate proper nouns in the web page with the appropriate concept from an ontology.</td>
<td>Yes</td>
</tr>
<tr>
<td>Armadillo</td>
<td>2004</td>
<td>Fully automatic</td>
<td>An initial URL.</td>
<td>The computer science department task.</td>
<td>Yes</td>
</tr>
<tr>
<td>KnowItALL</td>
<td>2004</td>
<td>Fully automatic, unsupervised</td>
<td>An extensible ontology and a small number of generic rule templates.</td>
<td>Building lists of named entities found on the Web.</td>
<td>Yes</td>
</tr>
<tr>
<td>Thresher</td>
<td>2005</td>
<td>Semi-automatic</td>
<td>Users highlight a few examples in a web browser.</td>
<td>Extract semantic content from sites with high degree DOM structure.</td>
<td>No</td>
</tr>
<tr>
<td>Textrunner</td>
<td>2007</td>
<td>Fully automatic, unsupervised</td>
<td>None</td>
<td>Open relation extraction from the Web.</td>
<td>No</td>
</tr>
<tr>
<td>Stat-Snowball</td>
<td>2009</td>
<td>Unsupervised</td>
<td>A set of seeds and an initial model. Assume that entities are given.</td>
<td>Identify relation tuples.</td>
<td>No</td>
</tr>
<tr>
<td>Cerno</td>
<td>2009</td>
<td>Semi-automatic</td>
<td>The tool is tuned to a particular domain by providing the system some hints, i.e., semantic clues, about concepts.</td>
<td>Entity recognition from accommodation ads and tourist board web pages.</td>
<td>Yes</td>
</tr>
<tr>
<td>GoNTogle</td>
<td>2010</td>
<td>Semi-automatic, supervised</td>
<td>User annotation history is used for model training.</td>
<td>Annotation of 500 papers according to an OWL ontology transformed from the ACM Computing Classification.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### C RESULTS

#### C.1 Named Entity Recognition

Table A.6. Some Notable Named Entity Recognition Systems, the Human Input Required, and their Reported Performance

<table>
<thead>
<tr>
<th>Systems</th>
<th>Scenario</th>
<th>Human Input</th>
<th>Performance</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armadillo (Dingli et al. 2003b)</td>
<td>Extracting personal data from Computer Science Department web sites.</td>
<td>A list of CS web sites.</td>
<td>P: 0.94, R: 0.37, F: 0.51</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>SemTag (Dill et al. 2003b)</td>
<td>Annotating a collection of approximately 264 million web pages, generating approximately 434 million automatically disambiguated semantic tags.</td>
<td>Knowledge base.</td>
<td>Accuracy: 0.82</td>
<td>Fully automatic</td>
</tr>
</tbody>
</table>

(Continued)
### Table A.6. Continued

<table>
<thead>
<tr>
<th>Systems</th>
<th>Scenario</th>
<th>Human Input</th>
<th>Performance</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>KnowItALL (Etzioni et al. 2005)</td>
<td>Building lists of named entities found on the Web.</td>
<td>A seed set of four examples.</td>
<td>P: 0.98, R: 0.76</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>Nadeau et al. (2006)</td>
<td>Named-entity recognition from the Internet, more than 50 entity types.</td>
<td>A seed of k manually generated entities for each entity type.</td>
<td>P: 0.88, R: 0.85, F1: 0.86</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>T-NER (Ritter et al. 2011)</td>
<td>Named entity recognition on Tweets.</td>
<td>Unlabeled Tweets, and Freebase as knowledge source.</td>
<td>P: 0.72, R: 0.60, F: 0.66</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>WebSets (Dalvi et al. 2012)</td>
<td>Extracting concept-instance pairs from an HTML corpus.</td>
<td>A corpus</td>
<td>Cluster purity: 0.83–0.99</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Mosallam et al. (2014)</td>
<td>Detecting person, location, and organization names in French.</td>
<td>An unlabeled corpus and a knowledge base.</td>
<td>P: 0.98, R: 0.85, F: 0.91</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>Ivanitskiy et al. (2016)</td>
<td>Detecting person, location, and organization names in Russian.</td>
<td>Six unlabeled Russian corpora.</td>
<td>(Person) P: 0.93, R: 0.86, F: 0.89</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Location) P: 0.93, R: 0.80, F: 0.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Organization) P: 0.80, R: 0.61, F: 0.68</td>
<td></td>
</tr>
</tbody>
</table>

For space reasons, only the best results in the optimal settings are presented here. The "P," "R," "F" stand for Precision, Recall, F1, respectively.

### C.2 Relation Extraction

### Table A.7. Some Notable Relation Extraction Systems, the Human Input Required, and their Reported Performance

<table>
<thead>
<tr>
<th>Systems</th>
<th>Scenario</th>
<th>Human Input</th>
<th>Performance</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIPRE (Brin 1999)</td>
<td>Extract (author, title) relation from the Web.</td>
<td>Five seed tuples of (author, title) pairs.</td>
<td>P: 0.95</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>Snowball</td>
<td>Extract (organization, location) relation.</td>
<td>Five seed tuples of (organization, location) pairs, and general regular expression.</td>
<td>P: 0.93</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>PANKOW (Cimiano et al. 2004)</td>
<td>Extract instance-concept relation in web pages with the appropriate concept from an ontology.</td>
<td>A web page and ontology.</td>
<td>F: 0.24</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>Textrunner (Banko et al. 2007)</td>
<td>Open relation extraction from the Web.</td>
<td>Without requiring any human input.</td>
<td>P: 0.84</td>
<td>Fully automatic, unsupervised (self-supervised)</td>
</tr>
<tr>
<td>StatSnowball (Zhu et al. 2009)</td>
<td>Identify relation tuples.</td>
<td>A set of seeds and an initial model. Assume that entities are given.</td>
<td>P: 0.79, R: 0.73, F: 0.76 (Sent500)</td>
<td>Unsupervised</td>
</tr>
</tbody>
</table>

(Continued)
Table A.7. Continued

<table>
<thead>
<tr>
<th>Systems</th>
<th>Scenario</th>
<th>Human Input</th>
<th>Performance</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gonzalez and Turmo (2009)</td>
<td>Relation mention detection.</td>
<td>An unannotated corpus, the year 2000 subset of the AQUAINT Corpus.</td>
<td>P: 0.51, R: 0.62, F: 0.56</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Bollegala et al. (2010)</td>
<td>Classifying relations in a social network.</td>
<td>Web pages downloaded from an online social network mining system.</td>
<td>P: 0.78, R: 0.52, F: 0.63</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Chen et al. (2017)</td>
<td>Chinese relation extraction.</td>
<td>A Chinese data set of Sina (a news portal).</td>
<td>P: 0.79</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Feldman and Rosenfeld (2006)</td>
<td>Extracting relations from the Web.</td>
<td>A short definition of the target relations.</td>
<td>Double or even triple of the recall achieved by KnowItAll for relatively rare relation instances.</td>
<td>Unsupervised</td>
</tr>
</tbody>
</table>

For space reasons, only the best results in the optimal settings are presented here. The “P,” “R,” “F” stand for Precision, Recall, F1, respectively.

C.3 Entity Linking

Table A.8. Some Notable Named Entity Linking Systems, the Human Input Required, and their Reported Performance

<table>
<thead>
<tr>
<th>Systems</th>
<th>Scenario</th>
<th>Human Input</th>
<th>Performance</th>
<th>Automation Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Hierarchy Fusion (Kamalloo and Rafiei 2018)</td>
<td>Toponym resolution.</td>
<td>No annotated corpus is used.</td>
<td>P: 0.81</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>XLink (Zhang et al. 2017)</td>
<td>Bilingual entity linking against Wikipedia and Baidu Baike.</td>
<td>No annotated corpus is used.</td>
<td>P: 0.91</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>WikIdRank (Fernández et al. 2012)</td>
<td>Entity linking task of the Knowledge Base Population track within the Text Analysis Conference.</td>
<td>No annotated corpus is used.</td>
<td>P: 0.77</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>VINCULUM (Ling et al. 2015)</td>
<td>General named entity linking.</td>
<td>No annotated corpus is used.</td>
<td>P: 0.79</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>LINDEN (Shen et al. 2012a)</td>
<td>General named entity linking.</td>
<td>No annotated corpus is used.</td>
<td>P: 0.94</td>
<td>Fully automatic</td>
</tr>
<tr>
<td>DBpedia Spotlight (Mendes et al. 2011)</td>
<td>Entity linking with DBpedia URIs</td>
<td>No annotated corpus is used.</td>
<td>P: 0.85</td>
<td>Fully automatic</td>
</tr>
</tbody>
</table>

For space reasons, only the best results in the optimal settings are presented here. The “P,” “R,” “F” stand for Precision, Recall, F1, respectively.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their helpful comments, in addition to Cees de Laat, Paul Martin, Jayachander Surbiryala, and ZeShun Shi for useful discussions.
REFERENCES

Mohamed Farouk Abdel Hady, Abubakrisedik Karali, Eslam Kamal, and Rania Ibrahim. 2014. Unsupervised active learning of
CRF model for cross-lingual named entity recognition. In Artificial Neural Networks in Pattern Recognition, Neamati

Saminda Abeyruwan, Ubbo Visser, Vance Lemmon, and Stephan Schürer. 2013. PrOntoLearn: Unsupervised lexico-
semantic ontology generation using probabilistic methods. In Uncertainty Reasoning for the Semantic Web II. Springer,
Berlin, 217–236.

336644

Alan Akbik, Larysa Visengriffyeva, Priska Herger, Holmer Hensen, and Alexander Löser. 2012. Unsupervised discovery of
relations and discriminative extraction patterns. Proceedings of COLING 2012, 17–32.

the International Conference on Web Intelligence (WI’17). ACM, New York, NY, 1138–1142. DOI: https://doi.org/10.1145/3106426.3109052

Anita Alicante, Anna Corazza, Francesco Isgrò, and Stefano Silvestri. 2016. Unsupervised entity and relation extraction
compmi.2016.01.014

Periklis Andritsos, Panayiotis Tspiras, Renée J. Miller, and Kenneth C. Sevcik. 2004. LIMBO: Scalable Clustering of Cate-
gorical Data. Springer, Berlin, 123–146. DOI: https://doi.org/10.1007/978-3-540-24741-8_9

Isabelle Augenstein, Andreas Vlachos, and Diana Maynard. 2015. Extracting relations between non-standard entities using
Processing, 747–757.

the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extrac-
tion from the web. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI’07). Morgan

distributional semantics. In Proceedingsof the 2015 Conference on Empirical Methods in Natural Language Processing,
499–504.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. A new form of web content that is meaningful

the 2006 SIAM International Conference on Data Mining. 47–58. DOI: https://doi.org/10.1137/1.9781611972764.5


(2003), 993–1022.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Embedding semantic relations into word repre-

Danushka Tarupathi Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: Unsupervised extraction of
semantic relations between entities on the web. In Proceedingsof the 19th International Conference on World Wide Web
(WWW’10). ACM, New York, NY, 151–160. DOI: https://doi.org/10.1145/1772690.1772707

Kalina Bontcheva and Hamish Cunningham. 2011. Semantic Annotations and Retrieval: Manual, Semiautomatic, and Auto-

COLING 2016, the 26th International Conference on Computational Linguistics. 2945–2955.

Svetla Boytcheva. 2018. Indirect association rules mining in clinical texts. In Artificial Intelligence: Methodology, Systems,
and Applications, Gennady Agichtein, Josef van Genabith, and Thierry Declerck (Eds.). Springer International Publishing,
Cham, 36–47.

International Workshop WebDB’98. Selected Papers, Paolo Atzeni, Alberto Mendelzon, and GianSalvatore Mecca (Eds.). Springer,
Berlin, 172–183. DOI: https://doi.org/10.1007/10704656_11

X. Liao and Z. Zhao
Unsupervised Approaches for Textual Semantic Annotation, A Survey


Unsupervised Approaches for Textual Semantic Annotation, A Survey


Edgar Gonzalez and Jordi Turmo. 2009. Unsupervised relation extraction by massive clustering. In 2009 Ninth IEEE International Conference on Data Mining, 782–787. DOI: https://doi.org/10.1109/ICDM.2009.81


Unsupervised Approaches for Textual Semantic Annotation, A Survey

66:41


Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. 2013b. Mining evidences for named entity disambiguation. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’13). ACM, New York, NY, 1079–1078. DOI: https://doi.org/10.1145/2487575.2487681


Hilário Oliveira, Rinaldo Lima, João Gomes, Rafael Ferreira, Fred Freitas, and Evandro Costa. 2012. A confidence-weighted metric for unsupervised ontology population from web texts. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7446 LNCS, 1 (2012), 176–190. DOI: https://doi.org/10.1007/978-3-642-32600-4_14


Saurav Sahay, Sougata Mukherjea, Eugene Agichtein, Ernest V. Garcia, Shamkant B. Navathe, and Ashwin Ram. 2008. Discovering semantic biomedical relations utilizing the web. ACM Transactions on Knowledge Discovery from Data (TKDD) 2, 1 (April 2008), Article 3, 15 pages. DOI: https://doi.org/10.1145/1342320.1342323


Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering 27, 2 (Feb. 2015), 443–460. DOI: https://doi.org/10.1109/TKDE.2014.2327028


Hassan A. Sleiman and Rafael Corchuelo. 2014. Trinity: On using trinary trees for unsupervised web data extraction. IEEE Transactions on Knowledge and Data Engineering 26, 6 (June 2014), 1544–1556. DOI: https://doi.org/10.1109/TKDE.2013.161


Received April 2018; revised March 2019; accepted April 2019