Relation Extraction Using Distant Supervision: A Survey

ALISA SMIRNOVA and PHILIPPE CUDRÉ-MAUROUX, eXascale Infolab,
University of Fribourg—Switzerland

Relation extraction is a subtask of information extraction where *semantic relationships* are extracted from natural language text and then classified. In essence, it allows us to acquire structured knowledge from unstructured text. In this article, we present a survey of relation extraction methods that leverage pre-existing structured or semi-structured data to guide the extraction process. We introduce a taxonomy of existing methods and describe distant supervision approaches in detail. We describe, in addition, the evaluation methodologies and the datasets commonly used for quality assessment. Finally, we give a high-level outlook on the field, highlighting open problems as well as the most promising research directions.

CCS Concepts: • **Information systems → Content analysis and feature selection**; **Information extraction**;

Additional Key Words and Phrases: Relation extraction, distant supervision, knowledge graph

ACM Reference format:
Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation Extraction Using Distant Supervision: A Survey. *ACM Comput. Surv.* 51, 5, Article 106 (November 2018), 35 pages.
https://doi.org/10.1145/3241741

1 INTRODUCTION

Despite the rise of semi-structured and structured data, text is still the most widespread content in companies and on the Web. As understanding text is deemed *AI-complete*, however, a popular idea is to extract structured data from text to make it machine-readable or processable. One of the main subtasks in this context is extracting semantic relations between various entities in free-form text, also known as *relation extraction*.

The classical survey on relation extraction [6] splits the approaches into two main branches: supervised and semi-supervised methods. Both families show significant limitations in the context of relation extraction, however. Supervised approaches [41, 43, 72] need annotated training data, which is expensive to produce. This limitation also makes supervised approaches hard to extend, in the sense that detecting new relation types requires new training data. In addition, supervised classifiers tend to be biased toward the domain of the text used for training and return suboptimal results on other textual contents. Semi-supervised approaches [1, 9, 17], however, typically rely on

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 683253/GraphInt).
Authors' addresses: A. Smirnova and P. Cudre-Mauroux, eXascale Infolab, University of Fribourg–Switzerland, Boullevard de Perolles 90, Fribourg, 1700, Switzerland; emails: {alisa.smirnova, philippe.cudre-mauroux}@unifr.ch.

0360-0300/2018/11-ART106 $15.00
https://doi.org/10.1145/3241741

bootstrap learning. They first use a small dataset to learn how to extract additional relations and then use the extracted relations for training in an iterative manner. Even though semi-supervised approaches significantly reduce the need of manual efforts to create training data, they still require an initial set of labeled pairs for every extracted relation. Thus, extending those approaches to extract new relation types typically involves human effort.

Unsupervised relation extraction techniques have also been proposed in the past [60] and further evolved into the subfield of Open Information Extraction [7, 18, 69]. Though unsupervised methods do not require any training data, they typically yield suboptimal results that are in addition hard to interpret and to map to an existing set of relations, schema or ontology [19]. This last drawback is a crucial shortcoming for many applications such as knowledge base refinement. In this survey, we do not discuss unsupervised approaches because of those limitations.

In recent years, an additional paradigm for automatic relation extraction was proposed, named *distant supervision*. Distant supervision combines the benefits of semi-supervised and unsupervised relation extraction approaches and leverages a *knowledge base* as a source of training data. The knowledge base that is exploited is typically storing data in a semi-structured fashion, using series of key-value pairs to create statements and to express information as *triples* in the form of *(subject, predicate, object)* connecting a given subject to an object using a predicate (e.g., *(Leo Tolstoy, wrote, War and Peace)*). The Resource Description Framework (RDF[1]) is a popular standard to express such triples. Note that the subject and object are customarily identified by unique identifiers in the knowledge base, and that they can be used interchangeably in the sense that the object of a given triple can also be the subject of further triples. Sets of such triples form a directed *graph* with nodes in the graph representing the subject and object in the triples and labeled edges representing the predicates connecting the subjects to the values. The subjects, objects, as well as the predicates used in such knowledge bases are often typed (by defining *classes* to classify subjects and objects and *properties* to classify predicates).

In this work, we survey the state of the art in relation extraction focusing on distant supervision approaches. Distant supervision is especially exciting today, as a number of large-scale knowledge bases, such as DBPedia[2] or Wikidata,[3] are readily available, and as numerous companies are building their own internal knowledge bases to power their business (see, for instance, Google[4] or Springer Natures's[5] knowledge graphs). Leveraging vast amounts of semi-structured data in generic or vertical domains thanks to such knowledge bases, distant supervision techniques can extract new triples from unstructured text with reasonable coverage and confidence. In the following, we classify distant supervision approaches into three main categories: noise reduction approaches, embeddings-based approaches, and approaches leveraging auxiliary information. We start our survey below by describing those three categories.

The rest of the article is structured as follows. We introduce in more detail the basic approach in relation extraction using distant supervision in Section 3. We review noise reduction methods in Section 4. Section 5 describes embeddings-based approaches including those leveraging convolutional neural networks. We discuss how to explore auxiliary information to automatically extract relations in Section 6. We also cover recent pieces of work using distant supervision for relation extraction in the biomedical domain in Section 7. We survey datasets, metrics, and experimental

[1] https://www.w3.org/RDF/.
[2] http://wiki.dbpedia.org/.
[3] https://www.wikidata.org/.
[4] https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html.
[5] https://www.springernature.com/cn/researchers/scigraph.

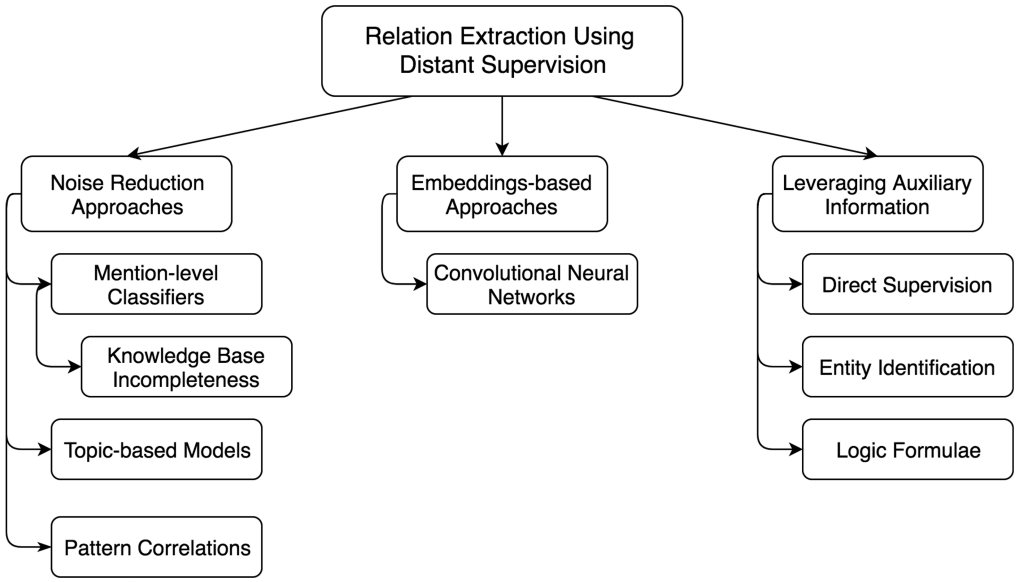

Fig. 1. Taxonomy of approaches.

results that have been used or produced by distant supervision approaches in Section 8. Finally, we comment on the various approaches and conclude in Section 9.

2 TAXONOMY OF APPROACHES

Figure 1 shows the taxonomy we introduce to classify the main distant supervision approaches. *Noise reduction* approaches focus on the core concepts behind distant supervision, when training data is not directly annotated and relation labels are automatically obtained from the knowledge base instead. While the knowledge base provides relation labels for a given entity pair, *mention-level classifiers* solve the task of predicting a relation expressed in a particular mention of the entity pair. A number of mention-level approaches tackle another problem inherent to distant supervision: data incompleteness. *Topic-based models* automatically identify both lexical and syntactic patterns that correspond to ontological relations. *Pattern correlation* methods study the textual representations of the relations and how to distinguish relation-specific patterns from more common patterns.

Embeddings-based approaches build vector representations of entities and relations and use those vectors for relation extraction. Deep learning approaches leveraging *Convolutional Neural Networks* are also based on word-embeddings and use pre-trained word vectors to encode input data. In addition, they take into account the specificities of distant supervision similar to mention-level approaches, i.e., the optimization task of the proposed models is defined with respect to the uncertainty of the instance labels.

The last category of approaches explores auxiliary information to improve relation extraction. Such information can come from manually annotated data (*Direct Supervision*), from additional information stored for the entities, such as entity types (*Entity Identification*) or from logical connections between relations (*Logic Formulae*).

We note that the listed categories are not mutually exclusive; for instance, most of the approaches leveraging auxiliary information are extensions of noise-reduction approaches. Some embeddings-based approaches also take into account noise reduction principles [73].

We describe these categories in more detail in Sections 4, 5, and 6.

3 DISTANT SUPERVISION PARADIGM

We start by describing the key ideas behind distant supervision for relation extraction. There is no strict definition of what is an entity and what is a relation. Typically, an *entity* is an instance of a "thing" that can be uniquely identified, while a *relation* is an association between entities [11]. By *entity*, we mean in the present context a proper noun, or one that could be the subject of an entry in Wikipedia or in a knowledge base. More formally, we define a relation as a triple (e_1, r, e_2), where e_1 and e_2 are two entities (subject and object or source and destination), r is a relation (a property) from a set R of all possible relations. In this survey, we only consider *closed relation extraction*, where the list of relations is known *a priori*. We also restrict our discussion to binary relations,[6] though we discuss potential research avenues for open relation extraction and non-binary relations in Section 9.

Given a text corpus $\mathcal{C}$ and a knowledge base $\mathcal{D}$, distant supervision matches relations from $\mathcal{D}$ to sentences from $\mathcal{C}$. More specifically, the idea is to first collect those sentences from the corpus $\mathcal{C}$ that mention entity pair (e_1, e_2), where both e_1 and e_2 exist in the knowledge base $\mathcal{D}$. If a sentence mentions (e_1, e_2) and there exists one triple (e_1, r, e_2) in the knowledge base, then the distant supervision approach *labels* this sentence as an instance (also called *mention*) of relation r. For example, the sentence

> South African entrepreneur **Elon Musk** is known for founding **Tesla Motors** and SpaceX.

mentions the tuple *(Elon Musk, Tesla Motors)*. Assuming the triple *(Elon Mask, created, Tesla Motors)* exists in the knowledge base,[7] the textual sentence is labeled with the *created* relation and can be used as training data for subsequent relation extractions. The set of the sentences that share the same entity pair is typically called a *bag* of sentences.

The idea of adapting a knowledge base to automatically generate training data for relation extraction was first proposed by Craven and Kumlien [12] for the biomedical domain (also known as *weak supervision*). The authors of Reference [68] developed a self-supervised classifier that uses information from Wikipedia pages and their infoboxes to predict article classes and to extract attribute values for infobox refinement.

Mintz et al. [40] generalized these two approaches. The authors formulated the *distant supervision assumption*:

Assumption 3.1. *"If two entities participate in a relation, then any sentence that contains those two entities might express that relation." [40]*

As we will see in the rest of this survey, a number of refinements of this basic assumption have been proposed. We describe the basic approach in more detail below.

3.1 Basic Approach

The relation extraction task can be considered as a multi-classification problem. That is, the goal is to predict, for every entity pair, the relation it participates in, picking the right relation from multiple classes. As for any standard classification task, the classifier needs training data where every entity pair is encoded into a *feature vector* and labeled with a corresponding relation. After

[6] The vast majority of the research papers considers binary relations only.
[7] See https://gate.d5.mpi-inf.mpg.de/webyago3spotlx/SvgBrowser?entity=%3CElon_Musk%3E&relation=%3Ccreated%3E for a concrete example.

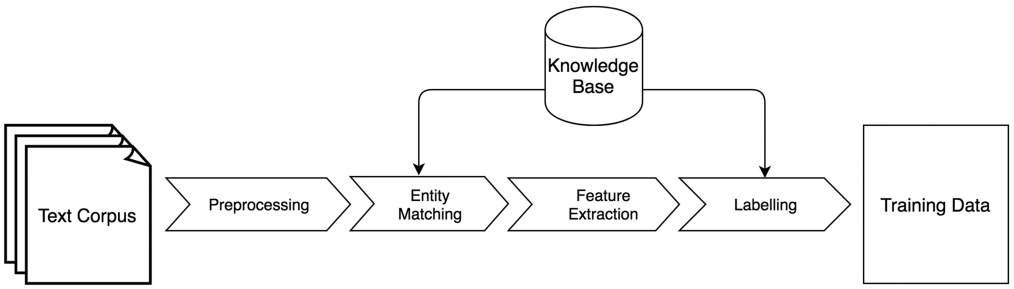

Fig. 2. Pipeline used for generating training data using distant supervision.

generating the training data, a test step is performed, where the classifier predicts relation labels for previously unseen entity pairs.

The basic process of transforming free text into training data using distant supervision is shown in Figure 2. The first step is preprocessing where classical Natural Language Processing tools are commonly used. Such tools include part-of-speech (POS) taggers (labelling words from the text using broad grammatical categories such as "noun" or "preposition"), dependency parsers (representing the syntactic structure of the sentence in terms of binary relations between tokens), and named entity recognizers (NER) (identifying potential entities from text).

The second step, Entity Matching, takes as input the textual entities identified in the previous step and tries to match them to one of the instances in the knowledge base (e.g., it tries to match "William Tell" as found in text to an instance in Wikidata, for example instance number "Q30908"[8]). This matching process is often complex, as the text snippet representing the entity (i.e., the *surface form* of the entity) can be different from the label of the same entity in the knowledge base (e.g., "Big Apple" vs. "New York City"), as words have to be disambiguated ("Apple" can stand for many different entities ranging from a piece of fruit to movies, rock bands or companies), and as text itself is often ambiguous and incomplete. Multiple effective methods were proposed in that context, leveraging word sense disambiguation [42], graph-based methods [23], or crowdsourcing [13].

At this point, whenever two entities are correctly matched to the knowledge base and co-occur in a given sentence, additional features get extracted from the sentence and the knowledge base to examine this case in more detail. Two types of features, lexical and syntactic, were originally proposed in Reference [40]. Very similar features were used in many subsequent approaches. Lexical features include the following:

- "The sequence of words between the two entities;
- The part-of-speech tags attached to these words;
- A flag indicating which entity name comes first in the sentence;
- A window of k words to the left of Entity 1 and their part-of-speech tags $k \in 0, 1, 2$;
- A window of k words to the right of Entity 2 and their part-of-speech tags $k \in 0, 1, 2$" [40].

Syntactic features are taken from the dependency parser. The output of the parser is a dependency tree where vertices represent words and edges represent syntactic relations between those words (see the example in Figure 3. Syntactic features commonly include the following:

- A dependency path between two entities. In Figure 3, the dependency path consists of three directed edges: (known, Elon Musk), (known, founding), (founding, Tesla Motors);

[8]See https://www.wikidata.org/wiki/Q30908.

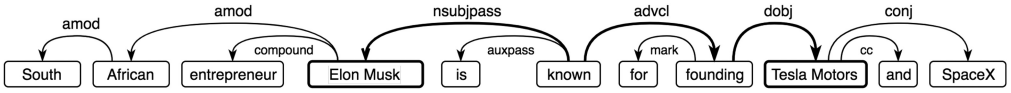

Fig. 3. Dependency parse tree with dependency path between "Elon Musk" and "Tesla Motors" highlighted in boldface.

- For each entity, a *window* node that is adjacent to one of the two entities but does not belong to the dependency path (the word "entrepreneur" in Figure 3).

Additional features can also be considered, including named entity tags (indicating whether the entities are persons, locations, or organizations), or various values attached to the entities in the knowledge base.

The final feature is a conjunction of the extracted sentence attributes. Two features match if and only if all their conjuncts match exactly. It is worth noting that this approach is suitable for large corpora only, since in a small corpus most of the features would appear infrequently (or not at all). It is common to combine the features from all mentions of the same entity pair into one feature vector representing this entity pair. The extracted features are subsequently added to the training data for the relation in question or considered as a potential relation instance in the test step.

The last step of this process is labelling, i.e., obtaining relation labels corresponding to the entity pair from the knowledge base. In general, there are two ways of accessing data in a knowledge base, either by using an API, which can be inefficient for high numbers of requests, or by downloading the latest dump of the knowledge base and querying it directly. Existing knowledge bases typically support both ways. Following the distant supervision assumption, *any* sentence mentioning a pair of entities that have a relation according to the knowledge base will be labeled with this relation. This heuristic is noise-prone, since the individual sentences can be erroneous labeled (i.e., when the pair of entities express a different relation from the one in the knowledge base). We discuss this shortcoming in the following sections. One of the specificities of distant supervision is that the knowledge base provides positive labels only. Thus, negative training data has to be produced synthetically, usually by sampling sentences containing entity pairs with no known relations in the knowledge base.

The test set is obtained similarly except from the labelling part, i.e., the first three steps (natural language processing, entity matching, and feature extraction) remain the same. After the training and test sets are constructed, a standard classifier can be applied for relation extraction. In Reference [40], the authors use a multi-class logistic classifier. In the test phase, the classifier can then be used to predict relation labels on the test set.

3.2 Strengths and Weaknesses

Supervised relation extraction approaches are specific to a certain vertical domain and often suffer from overfitting (as the corresponding training data is typically small and specific). Distant supervision, however, is applicable to large and heterogeneous corpora as it leverages large-scale knowledge defining semi-structured information for a large variety of entities.

Unfortunately, automatically labeled training data often includes noise, i.e., missing labels (false negatives) and wrong labels (false positives). On one hand, false negatives are mostly caused by incompleteness of the knowledge base. Two entities can be related in reality but their relation might be missing in the knowledge base, hence their mentions will be wrongly labeled as negative examples by distant supervision. On the other hand, two related entities may co-appear in a sentence that technically does not express their relation. Such examples might yield false positives using distant supervision. To illustrate this last point, we consider the following two examples:

Elon Musk is the co-founder, CEO, and Product Architect at **Tesla**.
Elon Musk says he is able to work up to 100 hours per week running **Tesla Motors**.

Both examples mention *Elon Musk* and *Tesla Motors* but only the first one indeed expresses the relation *co-founderOf*. Given a large and curated knowledge base, however, false positives are more likely to occur than false negatives (i.e., chances that two entities without a relation in the knowledge base are actually related in the real world are typically low).

In the next section, we turn to the category of distant supervision techniques tackling the problem of incorrect labels and noisy training data.

4 NOISE REDUCTION METHODS

We now describe a number of extensions of the basic distant supervision method that deal with noisy training data. In Reference [54], the following classification of noise reduction methods is introduced:

- *At-least-one models* reformulate Assumption 3.1 by stating that at least one sentence in the bag is a positive instance of the relation, but not necessarily all of them. In this work, we use the term *mention-level extraction* to emphasize the difference between two tasks: fact classification (whether entity pair has a relation) and sentence classification (whether a sentence expresses a relation);
- *Topic-based models* applied to relation extraction are based on the idea of separating distributions generated by patterns specific to the relation from the ones generated by the patterns specific to the entity pair or to the background text;
- *Pattern correlations models* construct a list of the patterns that are predicted not to express a target relation and use that list for reducing the number of wrong labels.

We adopt a similar classification in the following but also survey a number of important subcategories. We first present techniques that perform *mention-level relation extraction* in Section 4.1. Given a bag of sentences mentioning the same entity pair, these techniques classify each sentence as a positive or negative relation instance and use the results for assigning relation labels to the entity pair. Additionally, we discuss an important subclass of mention-level extractors and give an overview of approaches handling the incompleteness of the knowledge base and of the text corpus in Section 4.2. In Section 4.3, we introduce another class of approaches that focus on determining which textual patterns are indeed expressing a given relation. These approaches do not use the original distant supervision assumption. Instead, they distinguish different types of textual patterns by learning their probability distributions. Finally, we discuss approaches that directly exclude noisy textual patterns from the training data in Section 4.4.

4.1 Mention-Level Relation Extraction

The original distant supervision technique assumes that all sentences containing two related entities support the same relation between those two entities. This is often not the case in practice, particularly when the knowledge base is unrelated to the provided text corpus (e.g., when they cover two slightly different topics or come from two different sources). To obtain more accurate results with distant supervision, the authors of Reference [50] relax the original assumption and use the *expressed-at-least-once* assumption instead:

Assumption 4.1. *"If two entities participate in a relation, then at least one sentence that mentions those two entities might express that relation." [50]*

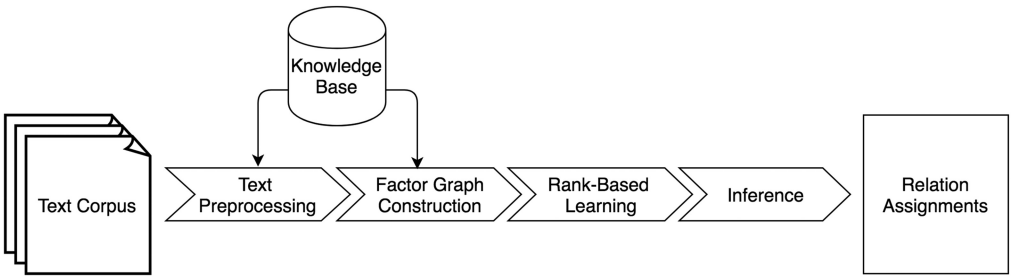

Fig. 4. The pipeline of the expressed-at-least-once approach.

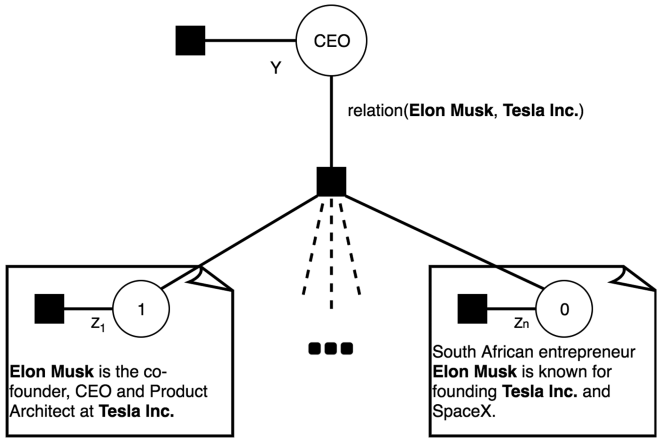

Fig. 5. The factor graph used for relation mention prediction and relation type identification (redrawn from Reference [50]). Circles denote variables, squares denote factors, where a factor is a function of its adjacent variables.

Under this relaxed assumption, the task of predicting relations becomes much more complicated, since we do not know which sentence is indeed expressing a relation both during training and testing. Thus, the labelling step described in Section 3.1 is no longer applicable. To tackle this problem, the authors of Reference [50] develop an undirected graphical model on top of a *factor graph*, a popular probabilistic graphical network representation where an undirected bipartite graph connects relation *variables* with *factors* representing their joint probabilities. Given two entities, the model predicts both a relation between them and a sentence expressing this relation.

The pipeline of this approach is shown in Figure 4. The text preprocessing step includes natural language processing, entity matching, and feature extraction as covered in Section 3.1. Each entity pair is encoded into a feature vector $\mathbf{x} = (x_1, x_2, \ldots x_n)$, where every component x_i includes additional information about mention i such as the dependency path between entities.

The factor graph consists of two types of variables that are connected with three types of factors (see Figure 5). The relation variable Y denotes a relation between two entities. The binary relation mention variable Z_i denotes whether a mention i expresses a given relation.

The authors model the conditional probability distribution $p(Y = y, \mathbf{Z} = \mathbf{z}|\mathbf{x})$ with respect to the following three factors:

- the relation factor $\Phi^r(y)$ that reflects the bias of the model toward the relation;
- the relation mention factor $\Phi^m(z_i, \mathbf{x}_i)$ that connects the components of the feature vector $\mathbf{x_i}$ and the assignment of Z_i;
- the factor that connects relation variables Y and their mentions $\Phi^{join}(y, \mathbf{z}, \mathbf{x})$.

All three factors are defined as functions of a parameter vector Θ and a binary feature function ϕ (see Reference [50] for details).

A Markov Chain Monte Carlo (MCMC) method is applied during the learning phase. A block Gibbs sampler randomly selects a relation mention i and jointly samples mention variable Z_i and the corresponding relation variable Y. The authors apply rank-based learning using the SampleRank framework [67], which allows to make parameter updates throughout the MCMC inference. At each step, SampleRank scores current assignments with two ranking functions. One of them measures the quality of the model (*model ranking*), while another one measures the agreement between the current assignment and the ground truth (*truth function*). If the rankings of two consecutive samples in the chain disagree (i.e., one of the functions increases while another one decreases), then SampleRank performs parameter update. A Gibbs sampler is also applied during the inference step to find the most likely configuration, i.e., the assignment $(y, \mathbf{z})$ maximizing the conditional probability $p(Y = y, \mathbf{Z} = \mathbf{z}|\mathbf{x})$.

This model is multi-instance single-label, i.e., it cannot capture the case where the same entity pair participates in multiple, different relations. In our above example, the relation between *Elon Musk* and *Tesla Motors* is not only *co-founderOf* but also *CEOOf*. The authors of References [25] and [61] propose undirected graphical models, called `MultiR` and `MIML − RE`, respectively, to perform *multi-instance multi-label* classification. Both models infer a relation expressed by a particular mention, thus the models are able to predict more than one relation between the two entities.

Let M be a bag, i.e., a set of sentences mentioning the same entity pair (e_1, e_2). The observed binary vector $\mathbf{Y} = \{y_i\}$ of size $|R|$ is assigned to a bag and denotes whether a relation $r_i(e_1, e_2)$ holds or not. Additionally, a latent vector $\mathbf{Z} = \{z_j\}$ of size $|M|$ is assigned to every mention in M. z_j takes one of the $|R|$ relation labels or a *NIL* label corresponding to the case when no relation is mentioned.

The `MultiR` model is a conditional probability model that learns a joint distribution of both mention-level and bag-level assignments. The conditional probability is defined in this context as follows:

$$p(\mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z}|\mathbf{x}; \theta) = \frac{1}{Z_x} \prod_r \Phi^{join}(y^r, \mathbf{z}) \prod_i \Phi^{extract}(z_i, x_i).$$

Here, $\mathbf{x} = \{x_i\}$ is a feature representation of mention i, while θ is a parameter of factor $\Phi^{extract}$.

The extraction factor $\Phi^{extract}$ can be interpreted as a weight of the current relation assignment to the sentence x_i. The factors Φ^{join} connect latent variables assigned to mentions with bag-level assignments:

$$\Phi^{join}(y^r, \mathbf{z}) = \begin{cases} 1 & \text{if } y^r = true \wedge \exists i : z_i = r \\ 0 & \text{otherwise.} \end{cases}$$

That is, relation r is predicted for a bag if and only if at least one of the sentences is predicted to express this relation.

The model learns a parameter vector $\Theta = \{\theta_j\}$ maximizing the conditional probability of the assignment. It is trained using a Perceptron-style additive parameter update scheme.

The `MIML − RE` model contains two layers of classifiers. The first-level classifier assigns a label to a particular mention. The second level has k binary classifiers, one for each relation, that classify entity pairs or bags. Given the predictions of the mention-level classifier, each bag-level classifier y_i determines if the relation r_i holds for a given pair of entities.

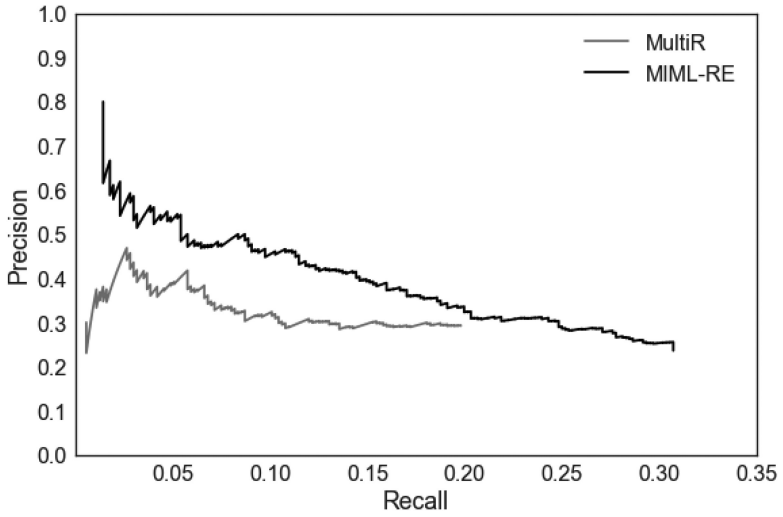

Fig. 6. Comparison between multi-instance multi-label classifiers.

The model is parameterized using the weights of the classifiers. The goal is to maximize the probability of both mention-level and bag-level assignments. Unfortunately, the log-likelihood for this model is not convex, therefore, the training algorithm maximizes the lower-bound of the likelihood. The authors used expectation maximization algorithm for this task.

We reproduced the experiments from Reference [61] using an open-source implementation of the models on a portion of the KBP dataset (see Section 8.1) provided by the authors.[9] The performance comparison between the two models is shown in Figure 6. From the figure, we observe that the `MIML − RE` model outperforms the `MultiR` model, especially on low-recall levels. Both `MultiR` and `MIML − RE` are widely used by further approaches. We present different optimizations of those models in Sections 4.2 and 6.

4.2 Data Incompleteness Handling

Distant supervision is specific in the sense that we obtain positive labels only and not negative ones from the knowledge base. The standard method to generate negative training data is to label mentions of unrelated entity pairs as negative samples. Since the knowledge base is incomplete, absence of any relation label does not imply absence of the corresponding relation. Hence, such a method potentially brings additional errors to the training data. In this section, we discuss multiple approaches that tackle this problem.

The authors of Reference [38] propose an additional layer to `MIML − RE` (see Section 4.1). The model input consists of a set of bags and binary vectors with values $\{Positive, Unlabeled\}$ of size $|R|$, reflecting the presence of relation r_i for a given entity pair in the knowledge base. An additional layer of latent variables l_i^r with values $\{Positive, Negative\}$ denote whether the relation r holds for a given bag i. This layer connects observed bag labels with the `MIML-RE` layers. The hyperparameter θ controls the fraction of the bags whose latent variable l_i^r is positive. As in Reference [61], the authors train their model using expectation maximization. Evaluation results show that the enhanced model predicts relation labels more accurately. Compared to `MIML − RE` and `MultiR`, it achieves better precision at almost all recall levels (see Table 1, `MIML − Semi` model).

[9] Code and data are available at http://nlp.stanford.edu/software/mimlre.shtml.

Table 1. Precision on Different Recall Levels, Held-out Evaluation, KBP Corpus

Recall	MultiR	MIML-RE	MIML-Semi
5%	39.1%	54.9%	**55.9%**
10%	30.3%	43.1%	**47.7%**
15%	30.8%	36.8%	**41%**
20%	30.2%	32.7%	**36.7%**
25%	—	**31.5%**	30.6%
30%	—	**24.9%**	23.8%

Similar to the previous approach, the authors of Reference [52] extend the `MultiR` model (see Section 4.1) to deal with missing labels. In distant supervision, the text corpus is aligned to the knowledge base while observing only the variable **d**, indicating whether the relation is present in the knowledge base. The authors add a variable **t** that denotes whether a relation fact is present in the text. The model penalizes disagreement between these two variables as follows:

$$\psi(t_i, d_i) = \begin{cases} -\alpha_{MIT} & \text{if } t_i = 0 \text{ and } d_i = 1 \\ -\alpha_{MID} & \text{if } t_i = 1 \text{ and } d_i = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The proposed model is quite flexible, since it allows us to assign different penalties to different relation facts depending on their probability to be missing from either the knowledge base or the text. For instance, certain facts are less likely to be absent from a data source, e.g., well-known facts such as "Elon Musk is the CEO of Tesla Inc." This intuition can be encoded by varying the entity-specific parameter $\alpha_{MID}^{(e_1,e_2)}$ proportionally to the popularity of given entities and the relation-specific parameter α_{MIT}^{r} proportionally to the number of true positives for relation r.

Another idea is to enhance the knowledge base, i.e., to infer entity pairs that are likely to participate in a relation, even if they are unlabeled in the knowledge base, and add them as positive samples. An approach based on this idea was developed by the authors of Reference [70] on top of the `MultiR` model. The authors create a passage retrieval system that ranks sentences in the original training corpus by their relevance to the specific relation. First, three sets of sentences are constructed for every relation r:

- $POS(r)$ consists of the sentences that mention any entity pair with relation r according to the knowledge base;
- $RAW(r)$ contains mentions of two entities of the corresponding types for relation r (e.g., person and organization for *founderOf* relation);
- $NEG(r)$ is the set of sentences that mention entity pairs (e_1, e_2) having their types compatible with relation r, while triple (e_1, r, e_2) is not present in the knowledge base but there exists some entity e_3 such that relation r holds between e_1 and e_3 (e.g., entity pair *(Elon Musk, Amazon)* for *founderOf* relation).

Then the authors train an SVM classifier using sentences in $POS(r)$ as relevant and sentences in $NEG(r)$ as irrelevant. The trained classifier estimates whether the sentence in $RAW(r)$ is relevant to the relation or not. This probability is also used as a ranking score for the sentence. Two types of lexical features are used for this task: Bag-Of-Words (set of tokens that do not include the entity mentions) and Word-Position (relative distance between a word and an entity mention). Pseudo-relevance feedback is then applied for each relation to rank entity pairs. Sentences with

higher relevance for a given relation imply a higher probability that the relation holds for the corresponding entity pair. Entity pairs whose average score is higher than some threshold are added to the knowledge base. Finally, the relation extraction model is trained on the updated knowledge base.

The authors provide a comparison of their model with the original MultiR model for the sentential extraction task (the task of predicting relation labels for one sentence). The authors report that their model yields consistently higher precision values than MultiR for similar recall levels. It achieves 78.5% precision and 59.2% recall at its peak performance, whereas MultiR achieves 72.4% precision and 51.9% recall only.

The authors of Reference [20] formulate the relation extraction task as a *matrix completion* problem. They take into account both erroneous labels and the incompleteness of the knowledge base. They describe three limitations in the training data:

- **Sparse features**, i.e., most of the extracted lexical and semantic features appear only once;
- **Noisy features** are caused by the distant supervision assumption;
- **Incomplete labels** are caused by knowledge base incompleteness.

As a consequence, the training matrix—whose rows correspond to entity pairs and columns correspond to the features—is deemed sparse.

Given two observed matrices $X_{train} \in \mathbb{R}^{n \times d}$ and $X_{test} \in \mathbb{R}^{m \times d}$ and a label matrix $Y_{train} \in \mathbb{R}^{n \times t}$, which is derived from the distant supervision assumption (where n is a number of training sentences, m is a number of testing sentences, d is a number of features and t is a number of possible relations) the task is to complete the matrix Z (i.e., to fill the unknown submatrix Y_{test}), defined as

$$Z = \begin{pmatrix} X_{train} & Y_{train} \\ X_{test} & Y_{test} \end{pmatrix},$$

under the assumption that the rank of the matrix Z is low. This assumption comes from the sparsity of the textual features and label incompleteness.

The authors proposed two optimization models called $\mathtt{DRMC-b}$ and $\mathtt{DRMC-1}$. As the matrix rank minimization problem is NP-hard, the authors adopt the fixed point continuation algorithm proposed in Reference [35] to find the optimum solution to their optimization problem. The authors compare their solution to a variety of other approaches, including MultiR and $\mathtt{MIML-RE}$, and report that their method consistently outperforms the baselines. We note that the performance of both models heavily depends on the hyperparameter controlling the rank of the underlying matrix: the lower the rank, the better the performance. One of the reasons behind the high performance of these approaches could stem from the sparsity of the data (which is a common problem for text classification tasks). The authors state that the proposed models effectively filter out noisy data while keeping enough information on the correlations between data points. Further experiments are needed to compare those results with the state of the art.

In Tables 1 and 2, we show comparative results for most of the models described in this section on two popular datasets: the *New York Times* corpus and the KBP corpus (see Section 8.1). The numbers were obtained from the corresponding papers. Here, we can see that the matrix completion approach performs significantly better than all other models. We also observe that both MultiR and $\mathtt{MIML-RE}$ take advantage of reducing erroneous negative labels.

4.3 Topic Models

Another widely used approach to improve text analysis and text classification is topic models. In this context, topics represent clusters of terms (words or patterns) that often co-occur in the documents together. The goal of topic models is to assign a topic to every term. More formally,

Table 2. Precision at Different Levels of Recall, Held-out Evaluation, NYT Corpus

Recall	**MultiR**	**MIML-RE**	**DRMC-b**	**DRMC-1**	**DNMAR**
5%	77%	66.7%	**96.8%**	95.9%	69%
10%	66.5%	61%	95%	**97%**	52%
15%	50.3%	45.9%	95.4%	**97%**	47.2%
20%	30.2%	33.7%	94.2%	**95.6%**	38.2%
25%	—	—	92%	**93.7%**	31%
30%	—	—	89.8%	**92.8%**	—

let us consider the conditional probability that word w belongs to topic t as $\phi_{wt} = p(t|w)$ and the conditional probability that document d belongs to topic t as $\theta_{td} = p(t|d)$. These probabilities form two matrices, Φ and Θ, respectively, which are the parameters of the topic model and have to be retrieved from the document collection. An expectation maximization algorithm can be used to obtain the parameters.

Topic models can be applied to distant supervision by mapping a *document d* to a sentence mentioning an entity pair, and a *topic t* to a relation. Sentences are represented by syntactic and lexical features, such as dependency paths between the entities or POS-tags. While the mention-level classifiers described above assign a relation to every sentence individually, the topic model classifiers are capable of capturing more general dependencies between textual patterns and relations, which can lead to improved performance.

Several approaches based on topic models were proposed for relation extraction. The authors of Reference [71] propose a series of generative probabilistic models in that context. Though the presented models are designed for unsupervised open relation extraction (i.e., the list of the target relations is not pre-specified), the authors also show how the detected clusters can improve distantly supervised relation extraction. All their proposed models are based on *latent Dirichlet allocation* (LDA), the topic model introduced in Reference [8]. The models differ in the set of features used and thus, in their ability to cluster patterns. A simple extension of LDA called Rel-LDA uses three features only: textual mentions of both entities and shortest dependency paths between them. Rel-LDA1 extends the set of features with *trigger* features (words from the dependency path except stop words, i.e., "known founding" in our example in Figure 3), lexical and POS tag patterns (the whole text between entities mentions and their POS-tags), NER tags of the entities, and the syntactic categories of the two entities. The Type-LDA model also takes into account entity types to yield a more accurate relation extraction.

The input of the models is represented as a set of *relation tuples*. Each tuple consists of three components: source entity, destination entity, and dependency path between those entities. The proposed models are applied to perform clustering on those tuples, i.e., the set of tuples is divided into subsets, where each subset (*cluster*) contains similar tuples. Hence, every tuple is associated with some clusterID. The authors follow the basic setting (see Section 3.1), though they extend the set of features with clusterID. On average, their best model achieves a 4.1% improvement over the basic distant supervision approach.

The advantage of generative topic models is that they are capable of "transferring" information from known patterns to unseen patterns, i.e., to associate a relation expressed by an already observed pattern to a new pattern. For instance, the Rel-LDA model correctly predicts relation *worksFor* for the previously unseen path "X, the anthropologist at Y," because this path has the same clusterID as the path "X, a professor at Y."

The authors of Reference [3] distinguish patterns that express a specific relation from the ones that are not relation-specific and can be observed across relations. For instance, pattern "was born in" is relation-specific while pattern "lived in" can be used across different relations, i.e., *mayorOf* or *placeOfDeath*. Their proposed models are also based on LDA. The submodels capture three subsets of patterns:

- "general patterns that appear for all relations;
- patterns that are specific for entity pairs and not generalisable across relations;
- patterns that are observed across most pairs with the same relation (i.e., relation-specific patterns)" [3].

Gibbs sampling is used to estimate the submodels from the input data. The authors explore two ways of extracting patterns: taking a text between the two entities (**Intertext**) and using a dependency path between them (**Syntactic**).

The models learn the probability that pattern w expresses relation r $P(r|w)$. As a baseline, the authors evaluate $P(r|w)$ as the number of times that pattern w connects two entities for which relation r holds divided by the total frequency of the pattern. Relation r is predicted to be true if the probability $P(r|w)$ is higher than some threshold p. Setting the threshold to $p = 0.5$, the proposed models are able to extract relations with a precision of 85% on average.

The authors make an important observation that relation extraction with distant supervision can be seen as a union of two distinct tasks: predicting the correct relation and finding sentence support, i.e., finding a sentence that expresses a particular relation. The difference is especially significant for such complicated relations as *nationality*: while the percentage of correctly extracted relations is 86%, the ratio of supporting sentences is only 34%. The reason behind this is that patterns such as "is a president of" are learnt to express *nationality*, since most of the presidents have indeed the nationality of the country where they govern. This example shows the specific difficulty of the relation extraction task, since some relations are strongly correlated and can be overlapping or partially imply each other, which is not the case for most of the other classification problems.

The authors of Reference [65] propose a similar approach. However, instead of generative topic models they apply diffusion wavelets to extract relation topics and subsequently integrate those topics into linear relation classifiers. Their approach can be split into three subsequent steps: data preparation and cleaning, extracting relation topics, and training relation classifiers. For this approach, the authors use a set of Wikipedia articles as text corpus and DBPedia as a knowledge base. Therefore, data construction relies on the heuristic that only the first sentence on the page that mentions entities is relevant for the relation between them. Each relation is represented with a set of rules extracted from the training data along with their popularity. A rule is a conjunction of entity pair types as well as a noun, verb and preposition from the shortest dependency path between the entities. To reduce the number of false positives, the authors remove instances that correspond to infrequent rules. This step produces 7,628 DBpedia relations. Some of these relations are heavily correlated (i.e., one relation can be a subclass of another, some relations are equivalent).

The next step is learning relation topics. First, each pair of relations is scored with a similarity function that basically measures the number of rules that the two relations share. Second, the correlation matrix is constructed and a technique called *diffusion wavelets* is applied to extract relation topics. The output is a matrix of lower dimensionality (this depends on the hierarchy levels), whose columns correspond to relation topics. This matrix is used to project an initial feature vector to the new vector space.

To perform classification, the authors construct a composite kernel and apply SVM classifiers. The final kernel is a linear combination of four kernels, three of them matching the argument

types, the dependency path, and the words outside the dependency path while the last one is a cosine similarity of the projections in the topic vector space.

In the evaluation results, the authors demonstrate that the number of relations decreases rapidly (from 7,628 relations in the beginning to 32 relation topics on the third level and one relation topic on the seventh level). Also, they demonstrate that these relation topics are interpretable: indeed, close relations are unified in one meaningful topic (i.e., at level 5 one of the three obtained topics consists of {birth_place, clubs, death_place, location}. This approach is corpus-specific. First, the knowledge base used is derived from the text corpus, thus, there is a bias toward better performance of the model. Second, the noise reduction approach relies on the nature of the used corpus (considering only the first sentence mentioning the entity pair).

The combination of hierarchical topic model and at-least-one learner is implemented in Reference [55]. As hierarchical topic model, the authors use a feature-based extension of the original hierarchical topic model [56]. As discriminative model, the authors use a perceptron model with an objective function that carries at-least-one assumption (Assumption 4.1) and assigns higher probability to a relation that exists in the knowledge base than to any other relation. To compensate for noisy extractions, the learner also gives higher probabilities to *NIL* labels (i.e., no relation) than to any relation that is not present in the knowledge base for a given entity pair. The resulting model combines the scores obtained by the underlying models. The authors compare two methods of obtaining the final score: a linear interpolation $S_{final} = 0.5 \cdot S_{topic} + 0.5 \cdot S_{perceptron}$ and the Pareto-frontier of the two-dimensional score vectors. Their experiments show that the combination model outperforms each individual model and that the simple linear interpolation works best. Compared to `MultiR` and $\texttt{MIML} - \texttt{RE}$, the combination of hierarchical topic model and perceptron model achieves slightly higher F1-score on the KBP corpus (0.307 against 0.242 for `MultiR` and 0.277 for $\texttt{MIML} - \texttt{RE}$). Hence, two distinct principles of noise reduction (at-least-once assumption and topic model) are able to improve on each other.

4.4 Pattern Correlations

The authors of Reference [62] propose to distinguish the patterns that express a relation from those that do not. Given some training data labeled with distant supervision and a per-relation list of the patterns that are detected as *not* expressing the corresponding relation, the authors reduce the number of erroneous labels from the training data in the following way: if a sentence contains a pattern from the negative list, then the corresponding label is removed. Thus, they reduce the number of wrong labels assigned by the distant supervision assumption. A multi-class logistic classifier analogous to Reference [40] is then trained on the filtered data.

The authors use generative models to predict whether a pattern expresses a relation. They model two probability distributions in this context. Let x_{rsi} be an observed binary variable denoting whether entity pair i appears in the text with pattern s and is labeled with relation r. A latent binary variable z_{rs} denotes whether pattern s expresses relation r. The probability that pattern s actually expresses relation r, $P(x_{rsi} = 1|z_{rs} = 1)$, is parameterized by a hyperparameter of the model a_r. The probability $P(x_{rsi} = 1|z_{rs} = 0)$ that pattern s does not express relation r is proportional to the number of entity pairs mentioned both with pattern s and with all correct patterns $t \neq s$:

$$P(x_{rsi} = 1|z_{rs} = 0) = a_r \frac{|(\bigcap_{\{t|z_{rt}=1, t\neq s\}} E_t) \bigcap E_s|}{|E_s|},$$

where E_t denotes the entity pairs that co-appear in the text with pattern t. The exact solution resulting from this model is computationally expensive. The authors propose an approximation instead, which requires a lower computational cost and less memory. Learning the parameters of the generative model is performed by maximizing the resulting log-likelihood.

The authors provide two sets of experiments. First, they measure the performance of their generative model on the task of predicting whether a pattern expresses the relation. Second, the authors compare the resulting model with the original classifier from Reference [40] and the MultiR model. The experiments demonstrate that filtering training data before learning is an efficient way to improve the model. The combination of logistic classifier and generative model for noisy pattern detection performs better than the original logistic classifier alone and achieves a precision comparable to the MultiR model on most recall levels.

The authors of Reference [39] propose several approaches for "label refinement." They develop a system for Knowledge Base Population (KBP).[10]

Their first technique called *pair selection* aims to measure on how strong is a correlation between the entities, i.e., how likely those entities are related. The obtained score is used to remove the ambiguous and noisy entity pairs from the training data. The second improvement generalizes the idea of relabeling the sentence with the most frequent relation that co-occurs with its dependency path. Instead of simple relabeling, a set of maximum entropy models is used to learn the weights of the features, thus making the relabeling process smoother. Two type of features are used to train these models: (i) conjunction of entity pair types, their order and the dependency paths between them, and (ii) conjunction of entity pair types, their order and the lexical paths between the entities. Third, the authors develop a set of hand-crafted and bootstrapped patterns to replace the labels obtained with distant supervision with the labels associated with these patterns. In addition, the authors merge similar relations into one class to make the dataset less imbalanced.

The evaluation results show that each of the proposed techniques helps in improving the performance of the system. However, it is worth emphasizing that a lot of manual effort was involved to produce the rules for label refinement.

In addition, a large collection of lexico-syntactic patterns called Freepal was created by the authors of Reference [30]. The authors perform named entity linking on the ClueWeb09 corpus,[11] filter it to remove spam, remove all sentences containing less than two entities, and finally parse it with an NLP toolkit. Then, patterns get extracted from the shortest dependency paths between two entities, as obtained using a dependency parser. The relation labels are associated with the patterns as per the distant supervision assumption (see Assumption 3.1). If the two entities participate in multiple relations, then the observed patterns are assigned to all possible relations. Additionally, every assignment receives a confidence score. To filter non-descriptive patterns, the authors measure the entropy H of each pattern using the likelihood estimates P for each observed relation Rel_i:

$$H(Pattern) = -\sum_i P(Rel_i) \log P(Rel_i).$$

Higher entropy values indicate that the pattern is observed in many relations, thus it is probably not very useful.

The authors point out the problem of the *long tail*. In the training data, they observe a power law distribution of relations in the corpus, i.e., there are few relations that are frequent while most of the other relations are infrequent. An analogous observation is made for patterns, i.e., some patterns are much more common than others. This indicates a potential difficulty when using distant supervision to extract less common lexico-syntactic patterns expressing a relation or to extract rare relations from the textual corpus.

[10] See https://tac.nist.gov/2012/KBP/index.html for more details.
[11] https://lemurproject.org/clueweb09/.

4.5 Discussion

One of the key issues in distant supervision is noisy labels. Some of the labels are false positives because of the distant supervision assumption, while some others are false negatives because of the incompleteness of the knowledge base. Mention-level classifiers and pattern learning methods can reduce the influence of incorrect positive labels. Another class of approaches is overcoming knowledge base incompleteness problems. Certain approaches, e.g., References [38, 52, 70], are based on mention-levels classifiers, hence, they reduce both types of incorrect labels.

5 EMBEDDINGS-BASED METHODS

The methods discussed above are based on a large variety of textual features. In this section, we introduce approaches that map the textual representation of relation and entity pair mentions onto a vector space. A mapping from discrete objects, such as words, to vectors of real number is called an embedding. The embeddings-based approaches for relation extraction do not require extensive feature engineering and natural language processing, but they still need Entity Matching tools (see Section 3.1) to connect the knowledge base entities with their textual mentions.

The authors of Reference [51] apply matrix factorization for the relation extraction task. The proposed models are able to extract relations defined in the knowledge base and, in addition, to predict surface patterns, e.g., *X visits Y* or *X is a scientist at Y*. The term "surface pattern" is used in Open Information Extraction and Question Answering in particular [16, 49]. In our context, surface patterns are considered as just one type of relations, thus covering the relations that are absent from the knowledge base.

The task is formulated as predicting the probability $p(y_{r,t} = 1)$ that relation r holds for entity pair t. Those probabilities form a matrix M with rows corresponding to entity pairs and columns corresponding to relations. The authors seek to find a low-rank factorization of M, such that

$$M \approx PR,$$

where P is a matrix of size $n \times k$, R is a matrix of size $k \times m$, n is the number of pairs, and m is the number of relations. Thus, P can be considered as a matrix of entity pairs embeddings and R is a relations embedding matrix. Relations come not only from the knowledge base but also from the union of the surface forms.

Subsequently, a set of models from exponential family were proposed that estimate the probability as a function of a real-vector parameter:

$$p(y_{r,t}|\theta_{r,t}) := \sigma(\theta_{r,t}) = (1 + \exp(-\theta_{r,t}))^{-1}.$$

The novelty of this approach is that the authors adopt a technique used by recommender systems, namely, *collaborative filtering*, to predict the probability $p(y_{r,t} = 1)$.

Different definitions of $\theta_{r,t}$ lead to different models. Four models are discussed:

(1) The latent feature model (F) defines $\theta_{r,t}$ as a measure of compatibility between entity pair t and relation r via the dot product of their embeddings.
(2) The neighborhood model (N) defines $\theta_{r,t}$ as a set of weights $w_{r,r'}$, corresponding to a directed association strength between relation r and r', where both relations are observed for tuple t.
(3) The entity model (E) also measures a compatibility between tuple t and relation r, but it provides latent feature vectors for every entity and every argument of relation r and thus can be used for n-ary relations. Entity types are implicitly embedded into the entity representation.

(4) The combined model (NFE) defines $\theta_{r,t}$ as a sum of the parameters defined by the three above models.

Stochastic Gradient Descent is used to maximize the objective, which assigns probabilities to the observed facts $f^+ := \langle r, t^+ \rangle \in O$ and the unobserved facts $f^- := \langle r, t^- \rangle \notin O$, so that $p(f^+) > p(f^-)$, i.e., the probability of the existing relation is higher than the probability of the relation that is not present in the knowledge base. Experiments show that the proposed models are more efficient in terms of time complexity and perform statistically significantly better than the baselines (i.e., the classifier based on Reference [40], the topic model from Reference [71], and the MIML − RE model). The combined NFE model yields a 63% mean average precision, while MIML − RE yields 56%.

Unlike the other approaches, matrix factorization models also extract surface patterns. Some of these patterns can be translated into relations in the knowledge base (e.g., "X was born in Y" or "X is a scientist at Y") while some cannot (e.g., "X visited Y").

The authors of Reference [10] extended RESCAL, a tensor decomposition approach described in Reference [44], by adding entity type information. The data is encoded as a tensor $\mathcal{X} \in \{0,1\}^{n \times n \times m}$, where n is the number of entities, m is the number of relations, $\mathcal{X}_{i,j,k} = 1$ if and only if a fact $r_k(e_i, e_j)$ is present in the knowledge base. RESCAL aims to find a rank-r approximation of a given tensor, where each slice $\mathcal{X}_k$ is factorized as

$$\mathcal{X}_k \approx \mathbf{A}\mathcal{R}_k\mathbf{A}^T,$$

where matrix R_k is relation-specific and matrix A remains the same for every relation.

The authors develop *Typed*-RESCAL or Trescal, which solves the same optimization problem as RESCAL but with respect to entity types. The presence of type information allows us to remove the entity-relation tuples of incompatible types from the training data, thus improving both the efficiency of the algorithm (by reducing the dimensionality of the matrices) and the prediction accuracy. The provided experimental results show that using Trescal alone for relation extraction is not very effective. According to the evaluation, Trescal achieves its best performance when combined with a matrix factorization approach [51]. The reason is that matrix factorization model does not share information between rows, i.e., between entity pairs, including the pairs containing the same entity. In Trescal each entity has its own vector representation that adds information about fine-grained entity types into the combined model.

The authors of Reference [66] propose an embedding model that is a composition of two submodels, one of which learns embeddings from text while the other one learns from the knowledge base. The textual submodel defines mention embeddings as $\mathbf{f}(m) = \mathbf{W}^T\Phi(m)$, where $\Phi(m)$ is a binary representation of mention m and $\mathbf{W}$ is a word embeddings matrix. A score function is a similarity function between a mention and a relation defined as

$$S_{m2r}(m, r) = \mathbf{f}(m)^T\mathbf{r}.$$

Stochastic Gradient Descent is used to learn $\mathbf{W}$ and $\mathbf{r}$ under the additional constraint that the ℓ_2 norm of each embedding vector is not greater than 1, minimizing the following objective:

$$\forall i, j, \quad \forall r' \neq r_i, r_j, \quad \max(0, (1 - S_{m2r}(m_i, r_i) + S(m_j, r')).$$

Thus, for every mention the model selects a relation that is more specific for particular mentions than other relations.

The knowledge base submodel uses the idea that vector embeddings $\mathbf{h}$ and $\mathbf{t}$—of the head entity h and tail entity t, respectively—and vector embedding $\mathbf{r}$ of relation r are connected by the approximation $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds, and $\mathbf{h} + \mathbf{r}$ is far from $\mathbf{t}$ when it does not hold. The score function measures the plausibility of the relation:

$$S_{kb}(h, r, t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2.$$

Similar to the textual submodel, the knowledge base submodel is trained using Stochastic Gradient Descent under the constraints reflecting the fact that the modified triple (with exactly one of the elements replaced with different values) should yield a lower score. The ℓ_2 norm of the embeddings is limited by 1 as well to prevent overfitting.

At test time, the predicted relation $\hat{r}_{h,t}$ for each entity pair (h, t) and a set of mentions $\mathcal{M}_{h,t}$ can be found as

$$\hat{r}_{h,t} = \arg\max_r \sum_{m \in \mathcal{M}_{h,t}} S_{m2r}(m, r).$$

If the predicted relation is not *NIL* (a marker denoting that there is no relation between the entities), then a composite score is defined as a sum of corresponding S_{m2r} and S_{kb} (on a test step S_{kb} is replaced with a calibration $\hat{S}_{kb}$).

The proposed approach predicts only one relation per entity pair, i.e., it is single-label. However, the model achieves results comparable with **MultiR** and **MIML − RE**, though the performance degrades rapidly when the recall value are > 0.1.

5.1 Neural Networks

The authors of Reference [73] perform relation extraction via Piecewise Convolutional Neural Networks (PCNNs). Convolutional Neural Networks (CNNs) are widely used for image classification as they are capable of capturing basic characteristics of an image, i.e., borders and curves, as well as capturing more complex structures, e.g., the number of legs on the image or the outline of objects. Similarly, CNNs are able to find specific n-grams in the text and classify relations based on these n-grams.

The learning procedure of PCNNs is similar to standard CNNs. It consists of four parts: *Vector Representation, Convolution, Piecewise Max Pooling*, and *Softmax Output*. In contrast to the embeddings-based approaches above, the proposed model does not build word embeddings itself. Instead, it uses a pre-trained Skip-gram model [36] to obtain word vectors. The skip-gram model learns word representations while predicting the context given the current word. Additionally, the proposed model encodes a position of the word in the sentence with *position features* introduced in Reference [74]. Position features denote the relative distance between a given word and both entities of interest e_1 and e_2. Two position embeddings matrices $\mathbf{PF}_1$ and $\mathbf{PF}_2$ are randomly initialized. The vector representation of a word is a concatenation of word embeddings and a position feature. The sentence is then encoded as a matrix $\mathbf{S} \in \mathbb{R}^{s \times d}$, where s is the length of the sentence and $d = d_w + 2 \cdot d_p$, where d_w is a word embedding size, and d_p is the size of the position embeddings.

The convolutional layer consists of n filters $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$. It scans the input matrix and produces the output matrix $\mathbf{C} \in \mathbb{R}^{n \times (s+w-1)}$, where w is the length of the filter. Each element c_{ij} of $\mathbf{C}$ is the result of the convolution operation between a vector of weight $\mathbf{w}_i$ and the w-gram of the input sequence:

$$c_{ij} = \mathbf{w}_i \mathbf{q}_{j-w+1:j} \quad 1 \le i \le n,$$

where $\mathbf{q}_{j-w+1:j}$ is the w-gram of the input matrix (that can be considered as a concatenation of the rows).

The pooling layer of the network is used to reduce the spatial size of the representation. In contrast to standard CNNs, in which only a single max pooling is applied, a piecewise max pooling procedure is applied. It computes the maximum value on each segment, where a segment is a portion of the text of the sentence: before e_1, between e_1 and e_2 or after e_2:

$$p_{ij} = \max(\mathbf{c}_{ij}) \quad 1 \le i \le n, \quad 1 \le j \le 3.$$

A non-linear function is then applied to a concatenation of the vectors $\mathbf{p}_i = \{p_{i1}, p_{i2}, p_{i3}\}$. The size of the output vector is fixed and depends on the number of filters only. A softmax classifier is then applied to the vector to obtain the confidence score of each relation.

An update procedure of network parameters is designed to overcome the wrong labels problem brought by distant supervision. Given all T training bags (M_i, y_i), where y_i is a binary vector denoting relation assignments to the corresponding bag, and a set of model parameters θ, which includes embedding matrices, filters and transformation matrix for the softmax classifier, the objective function is defined as follows:

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i | m_i^{j^*}; \theta),$$

where m_i^j denotes a mention in a bag M_i and j^* is defined as follows:

$$j^* = \arg\max_j p(y_i | m_i^j; \theta) \quad 1 \leq j \leq |M_i|.$$

The objective function is maximized using Stochastic Gradient Descent. A trained model predicts a relation for a bag (i.e., for an entity pair) if and only if a positive label is assigned to at least one entity mention.

A similar approach is proposed in Reference [34]. The authors explore different methods to overcome the wrong labelling problem. Let S be a bag of the sentences mentioning the same entity pair. Each mention x_i is represented by a vector $\mathbf{x_i}$, which is the output of convolutional and piecewise max pooling layers as described in the previous approach. The vector $\mathbf{s}$ corresponding to the bag is defined as a weighted sum:

$$\mathbf{s} = \sum_i \alpha_i \mathbf{x_i}.$$

Intuitively, weight measures how well the sentence x_i and a given relation r match. Two ways of defining the weights α_i are explored:

- Average, i.e., $\alpha_i = \frac{1}{|S|}$.
- Selective Attention, weight of the sentence depends on some function e_i, which scores compatibility between the sentence and relation:

$$\alpha_i = \frac{\exp(e_i)}{\sum_k \exp(e_k)}, \text{where} \quad e_i = x_i \mathbf{A} \mathbf{r_i}.$$

  Here, $\mathbf{A}$ is a diagonal matrix, and $\mathbf{r}$ is a relation embedding.

The authors hereby generalize the methods for selecting and weighing the relevant sentences for the multi-instance learning problem. Hence, the approach [73] discussed previously in this section is a special case of selective attention. The sentence with the highest probability gets the maximal score and other sentences get zero weight.

It is worth noting that both PCNN-based approaches perform single-label learning, i.e., only one relation is assigned to an entity pair. Nevertheless, evaluation results show that PCNNs outperforms feature-based approaches such as `MultiR` and `MIML − RE`. In Table 3, we provide a comparison between two CNN approaches and the `MIML − RE` model. Selective attention over sentences shows its effectiveness in solving the wrong labelling problem despite its simplicity. It turns the classification task into regression, i.e., selective attention predicts how strong the correlation between a sentence and a relation is instead of binary classifying whether a sentence supports a particular relation or not. This interpretation can simplify relation extraction especially for the distant supervision case.

Table 3. Precision on Different Recall Levels, Held-Out Evaluation, NYT Corpus

Recall	MIML-RE	PCNN	PCNN+ATT
5%	66.2%	75%	**77.4%**
10%	58.7%	64.9%	**69.2%**
15%	46.7%	55.1%	**64.2%**
20%	32.8%	50.8%	**60.7%**
25%	—	43.4%	**54.8%**
30%	—	40%	**51.4%**

5.2 Discussion

Building vector representations of words is a powerful approach for text analysis. This technique allows us to adopt algorithms from adjacent domains, such as recommender systems. In addition, pre-trained word embeddings models trained using a Skip-gram model proved to be useful for relation extraction. This can be explained through the built-in ability of the Skip-gram model to provide meaningful vectors that follow the following rule: $\mathbf{h_1} - \mathbf{t_1} \approx \mathbf{h_2} - \mathbf{t_2}$ whenever the relation between entity pairs (h_1, t_1) and (h_2, t_2) is the same.

Neural architectures model some of the features that are widely used for relation extraction (see Section 3.1). First, position features work similar to some lexical features, e.g., adjacent words. Second, piecewise max-pooling helps simulate lexical and syntactic features as it splits the sentence into three segments, thus modeling the feature "words in between" and roughly modeling dependency paths between the two entities. The authors of Reference [73] show that the same neural architecture but with single max-pooling performs similar to `MultiR` and `MIML − RE`.

The embeddings-based approaches do not require extensive natural language processing and feature engineering for relation extraction. However, they still need information about entity mentions that is commonly obtained through a named entity recognition tool.

6 LEVERAGING AUXILIARY INFORMATION FOR SUPERVISION

A number of distant supervision improvements use additional knowledge to enhance the model. Below, we review approaches that combine distant and direct supervision, i.e., approaches where the training data also includes sentences labeled manually. Then, we discuss approaches that enhance relation extraction by improving entity identification and exploring entity types. Finally, we discuss approaches that incorporate logical connections between relations (e.g., if two entities are related by the relation *fatherOf*, then logically these two entities cannot be related by *siblingOf*).

6.1 Direct Supervision

The authors of References [4, 47] study the impact of extending training data with manually annotated data, which is generally speaking more reliable than the labels obtained from the knowledge base.

In Reference [47], the authors emphasize that manually labeled data is more valuable than the data labeled automatically. Since the amount of hand-labeled data is relatively small, it is not effective to simply use this data during the standard training process. Instead, the authors perform feature selection on human labeled data and convert the results to the *training guidelines*. First, a supervised model is trained using hand-labeled data only. Then, the authors select an effective combination of three features:

- NER-tags of the two entities (e.g., person, organization, etc);
- a dependency path between the two entities;
- a span word, i.e., a word in the sentence between the two entities (optional).

This combination forms a *guideline* $g = \{g_i | i = 1, 2, 3\}$. Every guideline corresponds to a particular relation $r(g)$.

An additional layer of hidden variables h_{ij} is added to the MIML − RE model. If $\{x_{ij}\}$ is a feature representation of the mention j in the bag i and a latent variable z_{ij} denotes a relation expressed in this mention, then h_{ij} is defined as follows:

$$h_{ij}(x_{ij}, z_{ij}) = \begin{cases} r(g), & \text{if } \exists! \text{ guideline } g : g = \{g_k\} \subset \{x_{ij}\} \\ z_{ij} & \text{otherwise.} \end{cases}$$

Thus, this layer relabels a mention with a relation $r(g)$ if a feature representation of this mention contains one and the only guideline g. The rest of the model is the same as for MIML − RE. Experiments demonstrate that the proposed model performs consistently better that the original one, thus proving that even limited amount of manually annotated data can significantly improve the performance.

The approach described in Reference [4] is also an extension of the MIML − RE model. The authors do not amend the model itself; instead, they propose to initialize the model with manually annotated data, since the original model is sensitive to initialization (see Section 4.1). As manual annotation is very expensive, it is important to carefully select the sentences to annotate. In MIML − RE, a latent variable z denotes a relation expressed in a particular mention. The authors define a *measure of disagreement* for a specific z in the following way. k models are trained on k different subsets of data and each model predicts a multinomial distribution $P_c(z) = \{p_{c1}(z), \ldots, p_{cn}(z)\} : \sum_{i=1}^{n} p_{ci}(z = r_i) = 1$, where $n = |R|$ is the number of possible relations, p_i^c is the probability of relation r_i predicted by model c. The measure of disagreement is computed as a Jensen-Shannon divergence:

$$D(z) = \frac{1}{k} \sum_{c=1}^{k} KL(P_c(z) || P_{mean}(z)),$$

where $KL(P||Q)$ is the Kullback-Leiber divergence that measures the divergence between two distributions. The higher the divergence, the more the models disagree in determining a relation expressed in a particular sentence.

The authors study three selection criteria, i.e., how to select the sentences for manual annotation:

- sample uniformly (Uniform);
- take z with the highest disagreement (highJS);
- sample z with the highest disagreement (sampleJS).

The uniform criterion tends to select the sentences with low disagreement, since they make up the majority in the input data. Similarly, highJS selects noisy and non-representative examples, while SampleJS selects a mix of hard and representative examples, achieving the best performance.

The authors of Reference [4] provide a comparison of both extensions for the MIML − RE model that explore direct supervision, i.e., they compare the impact of proper initialization with the impact of adding guidelines. Table 4 illustrates the results of their evaluation including the original MIML − RE. Results show that a proper initialization allows to achieve slightly higher precisions at a given recall level ($\leq$0.2), while for higher recall levels both extensions perform similarly.

Table 4. Precision on Different Recall Levels, Held-Out Evaluation, KBP Corpus

Recall	MIML-RE	Pershina et al.	SampleJS
5%	58.4%	63.1%	**72.4%**
10%	50.3%	49.2%	**60.7%**
15%	41.9%	49.6%	**51.6%**
20%	32.8%	42%	**46.7%**
25%	30%	36.7%	**38.2%**
30%	25.8%	**33.1%**	32.7%

6.2 Entity Identification

In the previous sections, we focused on approaches that learn how relations between pair of entities are expressed in textual contents. There is, however, more than one way to mention the entities themselves. The relation extraction systems discussed above base their entity identification on NER tags and simple string matching. Thus, only entities mentioned by their canonical names or labels can be detected. However, it is common to use abbreviated mentions, acronyms or paraphrases. The task of connecting a textual mention with an entity in the knowledge base is called *entity linking*. We refer a reader to the survey on entity linking systems [59].

In addition, entity mentions can also be nouns (e.g., the profession of a person), noun phrases or even pronouns. Let us consider an example:

> Elon Musk is trying to redefine transportation on earth and in space. Through Tesla Motors—of which he is cofounder, CEO, and Chairman—he is aiming to bring fully electric vehicles to the mass market.

Here, the second sentence contains a relation mention (Elon Musk, cofounderOf, Tesla Motors) that cannot be extracted without co-reference resolution, which refers to the task of clustering mentions of the same entity together.

An additional way to make use of the entities themselves is to take into account their *types*. Entity types could be extremely useful for reducing noisy training data by simply filtering out entity pairs of the types not compatible with a given relation. Coarse entity types can be provided by named entity recognizers, which are used for entity identification by many approaches. Most frequent types are person, location and organization. Additionally, fine-grained types can be obtained from the knowledge base, such as *City* or *Scientist*.

In this section, we discuss the possibilities offered by entity identification methods and how these methods improve relation extraction.

The authors of Reference [5] explore a variety of strategies to make entity identification more robust. Moreover, they investigate the impact of co-reference resolution for relation extraction within more than one sentence.

The knowledge base used by distant supervision typically provides entity types and lexicalizations (i.e., less frequent names used to refer to an entity). Strategies for selecting, enhancing, or exploiting training data in this context are divided into three groups.

- Seed selection: detecting and discarding examples containing highly ambiguous lexicalisations;
- Relaxed setting: the distant supervision assumption is reformulated to perform relation extraction across sentence boundaries. It is defined as follows:

Assumption 6.1. *"If two entities participate in a relation, then any* paragraph *that contains those two entities might express that relation, even if not in the same sentence, provided that another sentence in the paragraph in itself contains a relationship for the same subject." [5]*

  - no co-reference resolution, the features requiring position of the subject are discarded;
  - co-reference resolution is performed by Stanford NLP co-reference resolution tool;
  - co-reference resolution implemented by the authors relies on synonym gazetteer and pronoun gazetteer (*gazetteer-based* method). The former is created from the set of synonyms, hypernyms and hyponyms of the class of the entity (e.g., "Book"), the latter is a set of the pronouns, including possessives, accordingly to the class gender or number.
- Information integration: aggregate information about the same entity pair from different documents.

The resulting model uses lexical features and named entity features and does not use any syntactic features. As in Reference [40], the authors train a multi-class logistic classifier in that context. Their approach is designed to perform relation extraction on very heterogeneous text corpora, such as those extracted from the Web. Thus, the dataset used for evaluation differs from most of the other approaches. To test their selection strategies, the authors built a corpus of around one million web pages by using the Google Search API with the query *"'subject entity' class_name relation_name,"* e.g., "'Tesla Motors' company created." For each entity, at most 10 pages were downloaded.

According to the evaluation results, the transition from sentence-level to paragraph-level relation extraction helps increase recall by up to 25%. Seed selection methods help remove ambiguities from training data and thus improve precision, but always at the cost of recall. The best model is a combination of the following models:

- a co-reference resolution model that uses synonym gazetteer;
- a seed selection model that discards lexicalisations that have high ambiguity with respect to the subject s. Ambiguity A_l^s of the lexicalisation l is the number of entities r that have the same lexicalisation when the triple (s, p, r) is present in the knowledge base for some relation p;
- a seed selection model that discards lexicalisations that are stop words (words that are highly frequent). The authors use a stop word list defined in Reference [33];
- a seed selection model that discards lexicalisations that have a high ambiguity with respect to the object o. Given the set of objects that share the same relation, ambiguity A_l^o is the number of objects with the same lexicalisation l. The ambiguity of each lexicalisation of an entity is considered as frequency distribution. The proposed model uses the 75th percentile of those frequency distributions as a threshold to discard ambiguous lexicalisations.

The authors of Reference [31] propose to use entity types and co-reference resolution for a more accurate relation extraction. Two sets of entity types are available: coarse types (PERSON, LOCATION, ORGANIZATION, and MISC) come from the named entity recognizer (NER), while fine-grained types come from the knowledge base [63]. To bring type-awareness into the system, the authors of the paper train independent relation extractors for each pair of coarse types, e.g., (PERSON, ORGANIZATION), (PERSON, LOCATION), and combine their predictions (Type Partitioning).

The classifier they use is an instance of the `MultiR` model (see Section 4.1). Candidate mentions with incompatible entity types are then discarded. Entity identification is carried out by using the Named Entity Linking (NEL) system from Wikipedia Miner [37]. For co-reference resolution, the authors use Stanford's sieve-based deterministic co-reference system [32]. Both NEL and co-reference resolution allow to extract more relation instances than just NER during training or

testing. Moreover, type constraints, i.e., filtering relation instances with arguments of incompatible types, helps precision at the expense of a small recall cost. Using NEL and co-reference resolution complicates the comparison with other models on a commonly used datasets, since those datasets contain only sentences mentioning two entities defined by NER. Therefore, the authors build an additional dataset named GoReCo to evaluate the influence of different entity identification approaches: named entity recognition, named entity linking alone and named entity linking with co-reference resolution.

Several models are trained on the KBP corpus and evaluated on the fully annotated GoReCo dataset. The best model that uses NEL with co-reference resolution during training and NEL with type constraints during extraction achieves a 18.5 F1-score while the model that uses only NER for both steps achieves a 11.6 F1-score.

Type-LDA (see Section 4.3) is a topic model considering fined-grained entity types based on latent Dirichlet allocation. In contrast to the previous approaches, it learns fine-grained entity types directly from the text, while having only NER tags. Trescal (see Section 5), finally, also uses information about entity types for relation extraction.

Robust entity identification and type-awareness are important parts of the relation extraction. First, robust entity identification provides improved accuracy on both training and testing data. Moreover, it allows to leverage fine-grained entity types stored in the knowledge base. Second, using type constraints allows to decrease computational costs. However, one should be aware that the knowledge base is typically incomplete, also when it comes to entity types.

6.3 Logical Formulae

In this section, we discuss approaches that consider logical connections or constraints between the relations. As an example, the relation *capitalOf* between two entities implies another relation *cityOf*. In addition, many relation pairs are mutually exclusive, e.g., *spouseOf* and *parentOf*.

The authors of Reference [53] explore in that context logical dependencies between relations. Their model is based on the matrix factorization approach from Reference [51] (see Section 5). The authors of the paper propose "to inject logical formulae into the embeddings of relations and entity pairs" [53]. They explore two methods: (i) pre-factorization inference and (ii) joint optimization. For pre-factorization, the model-inferred facts are added as additional training data. The drawback is that in this case only observed facts are involved and the dependencies between predicted facts are ignored. To overcome this problem the authors propose to optimize embeddings with respect to logical background knowledge. For that purpose the authors define loss function that includes both the probability of "*ground atoms*," i.e., relation facts, and the probability that given logical formula is true given the current model.

The authors focus on a particularly useful family of formulae: presence of one relation for some entity pair means that another relation also holds for this entity pair. For instance, the relation "city of" implies the relation "contained In." To extract those formulae, the authors run matrix factorization on the training data to learn the embeddings. Then, for each relation pair, they compute the score of the corresponding implication formula and manually filter the top 100 formulae.

According to the evaluation results, the proposed approach is capable to extract the relations that need to be added to the knowledge base while observing only textual patterns and extracted logic formulae. This situation arises in practice when a new relation is added to the knowledge base and no training data for this relation is available, thus, the extractor should rely on background knowledge only, e.g., logical formulae. On the complete dataset, joint optimization slightly outperforms the basic matrix factorization technique.

The approach described in Reference [22] is also based on the idea of using indirect supervision knowledge. The authors use in that sense:

- the consistency of relation labels, i.e., the inter-dependencies between relations that were mentioned previously (implications and mutual exclusions);
- the consistency between relation and arguments, i.e., entity type information retrieved from the knowledge base is used to filter inconsistent candidates;
- the consistency between neighbouring instances: the similarity function between two relation candidates is defined as the cosine similarity of the corresponding feature vectors. If mention m is labeled with relation r and mention m' is one of the k nearest neighbours of m, then mention m' is also labeled with relation r.

The authors use Markov Logic Network as a representation language.

6.4 Discussion

In this section, we discussed several possible ways to enhance the quality of distant supervision approaches by taking advantage of additional sources of knowledge, such as direct supervision, entity types information, robust entity identification or linking systems, and logical formulae connecting relations. Combining distant and direct supervision boosts the performance of the mention-level classifier. Using co-reference resolution allows us to extract more entity pairs and, therefore, increase the number of relation instances during the training step and relation candidates during the testing step, thus potentially improving recall. Robust entity linking and entity type awareness show their ability to improve precision of the mention-level classifiers. Additionally, the injection of logic formulae allows us to compensate for the lack of supervision. Using logic formulae allows us to enhance training data with labels that the knowledge base does not have.

7 APPLICATION TO BIOMEDICAL DOMAINS

Distant supervision has a long history in the biomedical domain. As approaches leveraging distant supervision in the biomedical domain somewhat evolved independently of other distant supervision techniques and consider different datasets, we review these approaches separately below.

Distant supervision was first applied to the biomedical domain by Reference [12] for extracting *subcellular-localization* relation. The technique helped obtain a much larger training dataset compared to manual annotation. A Naive Bayes classifier was subsequently trained to achieve results comparable with the one trained on hand-labeled data.

The authors of Reference [48] apply the `MultiR` model (see Section 4.1) for cancer pathway extraction. The model uses a set of lexical and syntactic features such as the sequence of words between two entities (in this scenario both entities are proteins) and both lexicalized and unlexicalized dependency paths between them. The authors perform two types of evaluation: on a small but fully annotated dataset called GENIA [28] (containing 800 training and 150 testing instances) and on a set of PubMed[12] abstracts where PID [58] was used as a knowledge base. The experiments on the GENIA dataset show that distant supervision performs comparably to fully supervised classifier achieving 25.6 F1 score, while the supervised model obtains 33.2. Manual evaluation on 300 extractions from the PubMed dataset demonstrates a high precision of the classifier (25%). This gives an estimation for the overall number of correct extractions of 372,000 and of 210,000 unique extracted triples, which is much larger that the number of relation triples obtained from PID (4,547 triples).

The authors of Reference [46] propose a novel approach for n-ary relation extraction across sentence boundaries and apply it for extracting drug-gene-mutation interactions. The approach presented in the paper is not based on existing relation extraction approaches. The authors

[12] https://www.ncbi.nlm.nih.gov/pubmed.

propose a novel architecture for relation extraction that uses graph LSTM Neural Networks. Graph LSTM is a generalisation of Long short-term memory networks [24] that have been successfully used for several NLP tasks. Traditionally, LSTMs process a sentence as a sequence of words. In this scenario, the only connection between two words is their adjacency. In contrast, graph LSTMs allow to handle other types of interactions between words such as syntactic dependencies. Thus, the input is represented as a *document graph* where nodes correspond to words and edges correspond to their dependencies. In this approach, the words are encoded with pre-trained word vectors similar to Reference [73]. We refer a reader to the original paper for further details on the neural architecture.

For their experiments, the authors use PubMed publications as a text corpus and two specific knowledge bases (GDKD [14] and CIVIC[13]). That gave around 140,000 positive training instances for drug-gene and drug-mutation binary relations. Automatic evaluation of binary relation extraction yields an accuracy of 76.7 for graph LSTMs while a CNN model [74] and a features-based model [48] achieve 74.9 and 75.2, respectively, for cross sentence extraction. We note that in this case the dataset is balanced, i.e., each class contains approximately the same number of training instances, which is not the case for the extraction on newswire corpuses. A manual evaluation of extracted triples (drug-gene-mutation) shows that given 59 positive triples from the knowledge bases the model is able to extract 1,461 triples with high confidence. The authors also reported that extraction across sentence boundaries gives twice as many binary relations than extraction from a single sentence (three to five times more for ternary relations).

8 DATASETS AND EVALUATION

In this section, we describe the most popular datasets used for evaluating relation extraction models and give an overview of the results achieved on them. We explain the basic techniques used for data preprocessing, then introduce the datasets used for evaluation. We conclude with an overview of experimental results. Table 5 summarizes the approaches discussed in this survey, along with the NLP tools, the knowledge bases, and the datasets used.

8.1 Datasets

To illustrate how much information could be potentially learned from textual data, we give a brief overview of the two largest Web corpora, ClueWeb09[14] and its successor ClueWeb12.[15] Both corpora consist of around 1 billion crawled Web pages. The English-language pages of both corpora have Freebase annotations [21]. In ClueWeb09, 75% of the annotated documents mention at least one Freebase entity and more than 5 billion entity mentions appear in the corpus. According to Reference [2], there are over 160 million sentences in ClueWeb09 that contain at least two entities that can be linked to Freebase. The authors of Reference [64] reported that a subset of 200 million sentences from ClueWeb12 mentions 98% of the entities from the knowledge base (a subset of Freebase[16]) producing 3.9 million relation instances.

Distant supervision techniques require a large text corpus for training purposes, since in a small dataset most of the features would only be found once [40]. The authors of References [40, 62, 65] evaluate their approach on Wikipedia pages. Since the knowledge base used was derived from Wikipedia articles, such results are skewed toward higher precision and recall values.

[13] https://civicdb.org/.
[14] http://lemurproject.org/clueweb09/index.php.
[15] http://lemurproject.org/clueweb12/index.php.
[16] https://www.microsoft.com/en-us/download/details.aspx?id=52312.

Table 5. Overview of distant supervision approaches

Paper	Name of the model	Type of the model	NLP tools	Knowledge base	Dataset
Mintz et al.		multi-class logistic classifier	Minipar[a]	Freebase	Wikipedia
Riedel et al.		undirected graphical model	Stanford NER, openNLP[b], MaltParser[c]	Freebase	NYT 2010 corpus
Hoffmann et al.	MultiR[d]	graphical model	openNLP, MaltParser	Freebase	NYT 2010 corpus
Surdeanu et al.	MIML-RE[e]	graphical model	Stanford CoreNLP	Freebase, Wikipedia infoboxes	NYT 2010 corpus, KBP corpus
Yao et al.	Rel-LDA, Rel-LDA1, Type-LDA	Latent Dirichlet Allocation	Stanford tools, MaltParser	Freebase	NYT 2010 corpus
Alfonseca et al.		hierarchical topic model	entity linking [37], inductive dependency parser [45]	Freebase	News articles
Roth et al.		combination of topic model and perceptron	Stanford CoreNLP	Wikipedia infoboxes	KBP corpus
Wang et al.		diffusion wavelets, kernels, SVM	slot grammar parser	DBPedia, YAGO	Wikipedia articles
Takamatsu et al.		generative model, multi-class logistic classifier	openNLP, MaltParser	Freebase	Wikipedia
Min et al.		label refinement, maximum entropy classifiers	Stanford parser[f]	Freebase	TAC KBP 2012 task
Xu et al.	IRMIE	passage retrieval combined with MultiR	Stanford NER, openNLP, MaltParser	Freebase	NYT 2010 corpus
Min et al.		MIML-RE with additional layer	Stanford CoreNLP	Wikipedia infoboxes	KBP corpus
Ritter et al.	DNMAR	extension of MultiR	Malt parser	Freebase	NYT 2010 corpus
Fan et al.	DRMC-1, DRMC-b	matrix completion	Stanford NER, openNLP, MaltParser	Freebase	NYT 2010, 2013 corpora
Riedel et al.		matrix factorization	NER tagging	Freebase	NYT 2013 corpus
Chang et al.	Trescal	tensor decomposition	NER tagging	Freebase	NYT 2013 corpus
Weston et al.		learning word embeddings	Stanford NER	Freebase	NYT 2010 corpus
Zeng et al.		Piecewise CNNs	Stanford NER	Freebase	NYT 2010 corpus
Lin et al.	PCNN-ATT	Piecewise CNNs with selective attention	Stanford NER	Freebase	NYT 2010 corpus
Pershina et al.	Guided DS	MIML-RE with additional layer	Stanford parser	Wikipedia infoboxes	KBP corpus
Angeli et al.		MIML-RE with careful initialisation	Stanford CoreNLP	Wikipedia infoboxes	KBP corpus
Augenstein et al.		multi-class logistic classifier	Stanford CoreNLP, Stanford coref	Freebase	Web pages
Koch et al.		MultiR[g]	Stanford NER, entity linking [37], Stanford coref, Stanford CoreNLP	Wikipedia infoboxes	TAC KBP 2009 task, GoReCo
Rocktäschel et al.		matrix factorization with logical formulae	NER tagging	Freebase	NYT 2013 corpus
Han et al.		MIML-RE, Markov Logic Network	Stanford CoreNLP	Wikipedia infoboxes	KBP corpus

[a] https://gate.ac.uk/releases/gate-7.0-build4195-ALL/doc/tao/splitch17.html.
[b] https://opennlp.apache.org/.
[c] http://www.maltparser.org/.
[d] code is available here: http://aiweb.cs.washington.edu/ai/raphaelh/mr/index.html.
[e] code is available here: http://nlp.stanford.edu/software/mimlre.shtml.
[f] https://nlp.stanford.edu/software/lex-parser.html.
[g] code is available here: http://homes.cs.washington.edu/ mkoch/re/.

The authors of Reference [50] evaluate their approach on the New York Times corpus [57]. The New York Times dataset contains over 1.8 million articles published between 1987 and 2007. The articles from 2005 to 2006 are used to construct training set whereas the articles from 2007 are used for testing set. We denote this dataset as NYT'10 in the rest of the section. The lexical and syntactic features are provided along with the relation labels obtained from Freebase.

For matrix factorisation approach [51], another dataset was constructed from the same text corpus (denoted as NYT'13 in the rest of this section). Here, the training set contains the articles from 1990 to 1999, whereas testing set contains the rest.

In Reference [61] the authors constructed another dataset, using resources for the 2010 and 2011 KBP shared tasks [26, 27] mostly. English Wikipedia infoboxes were used as a knowledge base. The document collection contains English Wikipedia dump from June 2010 along with 1.5 million documents taken from a large variety of sources, including blogs and newswire. Both the NYT corpus and the KBP corpus are widely used for evaluating relation extraction with distant supervision. One key characteristic of the NYT corpus lies in its structure. Since it was derived from news articles, it is skewed toward false positives, e.g., two entities are more likely to appear in the same sentence, because they share some context and not because the sentence expresses a relation. In contrast, a large part of KBP comes from Wikipedia articles that contain reference information about the entities, including many relations between them.

Additionally, the GoReCo dataset was developed by the authors of Reference [31] to evaluate relation extraction with co-reference resolution. They manually annotated the ACE 2004 newswire corpus, which contains 128 documents.

The authors of Reference [5] considered the task of extracting relation specifically from Web pages, therefore, they constructed a specific dataset containing around 1 million pages (see Section 6.2).

8.2 Preprocessing

A large variety of tools is used to perform natural language processing and to extract features from the text. As was mentioned in Section 3, feature-based relation extraction requires POS-tags, dependency parse trees and NER-tags. In addition, some of the approaches (see Section 6.2) require a named entity linker and a co-reference resolution tool. We list the preprocessing tools used in this context in Table 5.

Most of the approaches use Freebase[17] as the knowledge base.

8.3 Evaluation

The following three standard metrics are used to measure the algorithms performance:

- Precision $P = \frac{\text{Number of correctly extracted relations}}{\text{Total number of extracted relations}}$
- Recall $R = \frac{\text{Number of correctly extracted relations}}{\text{Actual number of relations}}$
- F-measure $F1 = \frac{2PR}{P+R}$

The researchers usually provide precision-recall curves, which is a well-known way of comparing different approaches in Information Retrieval.

With distant supervision one often does not have access to gold labels, but rather to labels from the knowledge base only, which is incomplete. Thus, the number of correctly extracted relations is underestimated (the model predicts relations that might not be covered by the knowledge base).

[17] https://developers.google.com/freebase/.

Moreover, the actual number of relation mentions is unknown; it can be roughly estimated as the number of relations in the knowledge base between the extracted pairs of entities.

Typically, the researchers perform two types of evaluation:

- held-out evaluations, where part of the knowledge base data is hidden during training and newly extracted relations are compared to this held-out data;
- manual evaluations, where small parts of the data is annotated by human evaluators usually using crowdsourcing.

Both methods present some disadvantages. In held-out evaluations, the fact that the knowledge base is incomplete affects the testing set, i.e., the extracted relations can be erroneously marked is incorrect. Thus, held-out evaluations underestimate the algorithms performance. However, manual evaluations provide more accurate results but are applicable for a relatively small subset of the data only.

Some of the approaches [25, 31, 70] also evaluate sentential extraction (see above Sections 4.1, 4.2, 6) on their models. The authors of Reference [25] perform their evaluation as follows. Let S^e be a set of sentences extracted by some system and S^F be the sentences that mention two arguments of a relation in the knowledge base. They sample 1,000 sentences from $S^e \cup S^F$ and manually label them with the correct relation label including *None* (no relation expressed). This evaluation scenario provides a reasonable approximation of the true precision but overestimates the actual recall, since a large set of sentences without any predicted relation is hidden during the evaluation process.

In addition, some authors report evaluation results on a subset of the most frequent relations. In this case, approaches measure the Average Precision of the relation as well as the Mean Average Precision to evaluate the model over the relations. For example, given the ranked list of the top 1,000 entity pairs, the authors of Reference [51] select the top 100 entity pairs and manually judge their relevance. The average precision is then calculated as the area under the precision-recall curve. The results show that some of the relations are "easier" to extract than others, i.e., precision (or average precision) of certain relations is substantially higher. For example, according to the evaluation of the matrix factorization approach [51], the most frequent relation *person/company* has an average precision of 0.79, while an average relation like *person/place_of_birth* has an average precision of 0.89 and rare relations (with only 1 or 2 mentions) have an average precision of either 1.0 or about zero.

9 CONCLUSIONS AND OUTLOOK

In this survey, we discussed the main relation extraction methods leveraging semi-structured data. We can summarize the two main challenges in distant supervision as follows:

(1) The labels automatically obtained from the knowledge base are noisy, since the sentences mentioning a pair of entities do not necessarily express a relation; moreover, different sentences mentioning the same entity pair can express different relations.
(2) The incompleteness of the knowledge base hurts both the training and the evaluation of the models.

The basic distant supervision approach (see Section 3.1) does not take into account those two problems and adopts a fairly ad hoc implementation leveraging a variety of hand-crafted features (both lexical and syntactic). However, even this basic approach shows that distant supervision is an efficient technique for processing large text corpora. It allows to achieve high precision on a complicated relation extraction task without any manual annotation of the given data.

To tackle the wrong labelling problem, various methods were proposed. Mention-level classifiers [25, 50, 61] assign relations to particular sentences. Then, based on these predicted relations, labels are assigned to the entity pairs. Two multi-instance multi-label models were proposed that are able to extract multiple relations for the same entity pair. Those models, called MultiR and MIML − RE, achieve similar performance on two widely used datasets (NYT and KBP). A number of further relation extraction approaches were built upon these models, as we discussed above in Sections 4.2 and 6.

Several approaches explored possible ways to better handle the fact that the knowledge base is incomplete, including most prominently various knowledge base refinement techniques applied before running the distant supervision pipeline [20, 38, 52, 70].

Pattern learning approaches [3, 55, 71] aim to better capture how relations are expressed in the text. For a given pattern and a relation, they attempt to learn the probability that the pattern is actually expressing a given relation. Pattern correlations approaches [62] model not-relevant patterns that can be abandoned from the training data.

Embeddings-based methods do not directly handle wrong labels. They build meaningful word projections onto vectors spaces and use those embeddings instead of hand-crafted lexical and syntactic features. CNN-based approaches apply convolutional neural networks that transform the input data into 3D tensors where each word is encoded with word embeddings and position features. Though CNN-based approaches do not perform multi-label relation extraction, they achieve higher results than traditional feature-based methods.

Finally, a set of approaches explore the ability of improving existing methods by leveraging technique to process additional sources of information such as selective direct supervision, robust entity identification, and logical formulae inclusion. Selective direct supervision helps to improve the accuracy of mention-level classifiers with little manual intervention. Robust entity identification enriches training data for relation extraction with fine-grained entity types and new relation instances obtained using co-reference resolution. Injecting logical background knowledge is a novel direction in relation extraction with distant supervision that has been shown to be effective even for scenarios where no relation labels are available.

We observe that embeddings-based approaches gained more popularity in the recent years. The reason behind this popularity stems from the fact that continuous representations act like a bridge between text processing and recent advances in mathematical (e.g., matrix factorisation) and machine learning techniques (e.g., neural networks). How to produce high-quality embeddings leveraging NLP techniques is still largely an open question, however.

Given recent advances, we are expecting more industrial systems to include distant supervision components in the near future. One recent example of that trend is the DeepDive project [75], which leverages distant supervision and which was applied to several domains before being acquired by a large company.

As for future work in this field, we identify several possible avenues. Distant supervision is a key technology for automatic relation extraction from big corpora and noisy labels remain a significant shortcoming of this method. A novel direction to tackle this problem is to involve crowdsourcing or human-in-the-loop methods to drastically reduce noisy labels. The idea is to try to automatically identify noisy labels and then to ask anonymous human annotators to manually review them. Two important questions arise in that context: (i) How do we select the labels to be reviewed to minimize human labor while maximizing the system's performance? (ii) How do we propagate the feedback received from human annotators to further labels to refine as much training data as possible? Though one possible answer to the first question is proposed in Reference [4], more research in that area is called for.

We also identify several directions for extending existing relation extraction techniques. One is to take into account more context and background knowledge for relation extraction. One option in that context would be to consider more than just one sentence (i.e., the whole paragraph) when extracting relations analogously to the relaxed setting in Reference [5]. Another promising direction would be to investigate further sources of background knowledge, including for instance external sources such as specialized sources for more accurate extraction on specific relation types (e.g., geographical databases or specialized magazines for geographical relations). N-ary relation extraction also remains an active research area [15, 46].

Another key open problem, mentioned in Reference [30], is the *long tail* of relations, i.e., the fact that there are a few relations that are frequent while most of the relations are infrequent. Improving distant supervision techniques beyond the top-100 most frequent relations is a formidable challenge. While knowledge bases are likely to contain a small number of facts related to infrequent relations only, unsupervised pattern clustering [71] and logical background knowledge [22, 53] could help in that sense. The authors of References [51, 53] address an adjacent problem, namely, extracting not only relations from the knowledge base but also surface patterns that represent a broader class of facts.

Deep learning is another exciting technique that is deemed to impact relation extraction in the future. At this point however, there are only a few research papers leveraging recurrent neural networks, though CNNs are widely used for sentence classification and relation extraction [29, 43, 73].

ACKNOWLEDGMENTS

We gratefully thank Julien Audiffren as well as the anonymous reviewers for their comments on earlier versions of this manuscript.

REFERENCES

[1] Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries*. 85–94.

[2] Alan Akbik, Thilo Michael, and Christoph Boden. 2014. Exploratory relation extraction in large text corpora. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 2087–2096.

[3] Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. 2012. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. 54–59.

[4] Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1556–1567.

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

[6] Nguyen Bach and Sameer Badaskar. 2007. A survey on relation extraction. Language Technologies Institute, Carnegie Mellon University.

[7] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, Vol. 7. 2670–2676.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[9] Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of the International Workshop on the World Wide Web and Databases (WebDB'98)*. Springer, 172–183.

[10] Kai-Wei Chang, Scott Wen-tau Yih, Bishan Yang, and Chris Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1568–1579.

[11] Peter Pin-Shan Chen. 1976. The entity-relationship model—Toward a unified view of data. *ACM Trans. Database Syst.* 1 (1976), 9–36.

[12] Mark Craven, Johan Kumlien, et al. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*. 77–86.

[13] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st World Wide Web Conference (WWW'12)*. 469–478.

[14] Rodrigo Dienstmann, In Sock Jang, Brian Bot, Stephen Friend, and Justin Guinney. 2015. Database of genomic biomarkers for cancer drugs and clinical targetability in solid tumors. *Cancer Discov.* 5 (2015), 118–123.

[15] Patrick Ernst, Amy Siu, and Gerhard Weikum. 2018. HighLife: Higher-arity fact harvesting. In *Proceedings of the World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1013–1022.

[16] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM* 51 (2008), 68–74.

[17] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artific. Intell.* 165, 1 (2005), 91–134.

[18] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, vol. 11. 3–10.

[19] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. 1535–1545.

[20] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y. Chang. 2014. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14)*. 839–849.

[21] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora, Version 1. *Retrieved from* http://lemurproject.org/clueweb09/FACC1/Cited by.

[22] Xianpei Han and Le Sun. 2016. Global distant supervision for relation extraction. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 2950–2956.

[23] Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. 765–774.

[24] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9 (1997), 1735–1780.

[25] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1*. 541–550.

[26] Heng Ji, Ralph Grishman, and Hoa Dang. 2011. Overview of the TAC2011 knowledge base population track. In *Proceedings of the Text Analysis Conference*.

[27] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Text Analysis Conference*.

[28] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Biomedical Natural Language Processing Workshop Companion Volume for Shared Task (BioNLP@HLT-NAACL'09)*. 1–9.

[29] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1746–1751.

[30] Johannes Kirschnick, Alan Akbik, and Holmer Hemsen. 2014. Freepal: A large collection of deep lexico-syntactic patterns for relation extraction. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. 2071–2075.

[31] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1891–1901.

[32] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Comput. Linguist.* 39 (2013), 885–916.

[33] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.* 5 (2004), 361–397.

[34] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*.

[35] Shiqian Ma, Donald Goldfarb, and Lifeng Chen. 2011. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* 128 (2011), 321–353.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv Preprint arXiv:1301.3781*.

[37] David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*. 509–518.

[38] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the Human Language Technologies Conference of the North American Chapter of the Association of Computational Linguistics*. 777–782.

[39] Bonan Min, Xiang Li, Ralph Grishman, and Ang Sun. 2012. New york university 2012 system for KBP slot filling. In *Proceedings of the 5th Text Analysis Conference (TAC'12)*.

[40] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 1003–1011.

[41] Raymond J. Mooney and Razvan C. Bunescu. 2006. Subsequence kernels for relation extraction. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'05)*. 171–178.

[42] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: A unified approach. *Trans. Assoc. Comput. Linguist.* 2 (2014), 231–244.

[43] Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing (VS@NAACL-HLT'15)*. 39–48.

[44] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. 809–816.

[45] Joakim Nivre. 2006. *Inductive Dependency Parsing*, Vol. 34. Springer.

[46] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence N-ary relation extraction with graph LSTMs. *Trans. Assoc. Comput. Linguist.* (2017). arXiv preprint arXiv:1708.03743.

[47] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. 2014. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL'14)*. 732–738.

[48] Hoifung Poon, Kristina Toutanova, and Chris Quirk. 2015. Distant supervision for cancer pathway extraction from text. In *Proceedings of the Pacific Symposium on Biocomputing*. 120–131.

[49] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 41–47.

[50] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'10)*. Springer, 148–163.

[51] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the Human Language Technologies Conference of the North American Chapter of the Association of Computational Linguistics*. 74–84.

[52] Alan Ritter, Luke Zettlemoyer, Oren Etzioni, et al. 2013. Modeling missing data in distant supervision for information extraction. *Trans. Assoc. Comput. Linguis.* 1, 367–378.

[53] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT'15)*. 1119–1129.

[54] Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC@CIKM'13)*. 73–78.

[55] Benjamin Roth and Dietrich Klakow. 2013. Combining generative and discriminative model scores for distant supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. 24–29.

[56] Benjamin Roth and Dietrich Klakow. 2013. Feature-based models for improving the quality of noisy training data for relation extraction. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM'13)*. 1181–1184.

[57] Evan Sandhaus. 2008. The new york times annotated corpus. *Proceedings of the Linguistic Data Consortium*.

[58] Carl F. Schaefer, Kira Anthony, Shiva Krupa, Jeffrey Buchoff, Matthew Day, Timo Hannay, and Kenneth H. Buetow. 2009. PID: The pathway interaction database. *Nucleic Acids Res.* 37 (2009), 674–679.

[59] Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* 27 (2015), 443–460.

[60] Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 304–311.

[61] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 455–465.

[62] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. 721–729.

[63] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. 2013. Trank: Ranking entity types using the web of data. In *Proceedings of the 12th International Semantic Web Conference on the Semantic Web (ISWC'13)*. Springer, 640–656.

[64] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1499–1509.

[65] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. 2011. Relation extraction with relation topics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. 1426–1436.

[66] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. 1366–1371.

[67] Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. 2009. Samplerank: Learning preferences from atomic gradients. In *Proceedings of the Workshop on Advances in Ranking: Neural Information Processing Systems (NIPS'09)*.

[68] Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM'07)*. 41–50.

[69] Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*. 118–127.

[70] Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*. 665–670.

[71] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1456–1466.

[72] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.* 3 (2003), 1083–1106.

[73] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. 1753–1762.

[74] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 2335–2344.

[75] Ce Zhang. 2015. DeepDive: A data management system for automatic knowledge base construction. University of Wisconsin-Madison, Madison, Wisconsin.

Received March 2018; revised July 2018; accepted July 2018