Knowledge Base Population using Semantic Label Propagation

Lucas Sterckx, Thomas Demeester, Johannes Deleu, Chris Develder

Ghent University – iMinds
Technologiepark Zwijnaarde 15, BE-9052 Ghent, Belgium

Abstract

Training relation extractors for the purpose of automated knowledge base population requires the availability of sufficient training data. The amount of manual labeling can be significantly reduced by applying distant supervision, which generates training data by aligning large text corpora with existing knowledge bases. This typically results in a highly noisy training set, where many training sentences do not express the intended relation. In this paper, we propose to combine distant supervision with minimal human supervision by annotating features (in particular shortest dependency paths) rather than complete relation instances. Such feature labeling eliminates noise from the initial training set, resulting in a significant increase of precision at the expense of recall. We further improve on this approach by introducing the Semantic Label Propagation (SLP) method, which uses the similarity between low-dimensional representations of candidate training instances to again extend the (filtered) training set in order to increase recall while maintaining high precision. Our strategy is evaluated on an established test collection designed for knowledge base population (KBP) from the TAC KBP English slot filling task. The experimental results show that SLP leads to substantial performance gains when compared to existing approaches while requiring an almost negligible human annotation effort.

Keywords: Relation Extraction, Knowledge Base Population, Distant Supervision, Active Learning, Semi-supervised learning

1. Introduction

In recent years we have seen significant advances in the creation of large-scale knowledge bases (KBs), databases containing millions of facts about persons, organizations, events, products, etc. Examples include

*Corresponding author
Email address: lucas.sterckx@intec.ugent.be (Lucas Sterckx)

Wikipedia-based KBs (e.g., YAGO [1], DBpedia [2], and Freebase [3]), KBs generated from Web documents (e.g., NELL [4], PROSPERA[5]), or open information extraction approaches (e.g., TextRunner [6], PRISMATIC [7]). Other knowledge bases like ConceptNet [8] or SenticNet [9] collect conceptual information conveyed by natural language and make them easily accessible for systems performing tasks like commonsense reasoning and sentiment analysis[10]. Besides the academic projects, several commercial projects were initiated by major corporations like Microsoft (Satori[1]), Google (Knowledge Graph [11]), Facebook[2], Walmart [12] and others. This is driven by a wide variety of applications for which KBs are increasingly found to be essential, e.g., digital assistants, or for enhancing search engine results with semantic search information.

Because KBs are often manually constructed, they tend to be incomplete. For example, 78.5% of *persons* in Freebase have no known *nationality* [13]. To complete a KB we need a knowledge base population (KBP) system that extracts information from various sources of which a large fraction comprises unstructured written text items [11]. A vital component of a KBP system is a relation extractor to populate a target field of the KB with facts extracted from natural language. Relation extraction (RE) is the task of assigning a semantic relationship between (pairs of) entities in text.

There are two categories of RE systems: (i) *closed*-schema IE systems extract relations from a fixed schema or for a closed set of relations while (ii) *open* domain IE systems extract relations defined by arbitrary phrases between arguments. We focus on the completion of KBs with a fixed schema, i.e., closed IE systems.

Effective approaches for closed schema RE apply some form of supervised or semi-supervised learning [14, 15, 16, 17, 18, 19] and generally follow three steps: (i) sentences expressing relations are transformed to a data representation, e.g., vectors are constructed to be used in feature-based methods, (ii) a binary or multi-class classifier is trained from positive and negative instances, and (iii) the model is then applied to new or unseen instances. To review the evolution of these and other natural language processing techniques readers can refer to the article by Cambria and White [20].

Supervised systems are limited by the availability of expensive training data. To counter this problem, the technique of iterative bootstrapping has been proposed [21, 22] in which an initial seed set of known facts is used to learn patterns, which in turn are used to learn new facts and incrementally extend the training set. These bootstrapping approaches suffer from semantic drift and are highly dependent on the initial seed

[1] https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing

[2] http://www.insidefacebook.com/2013/01/14/

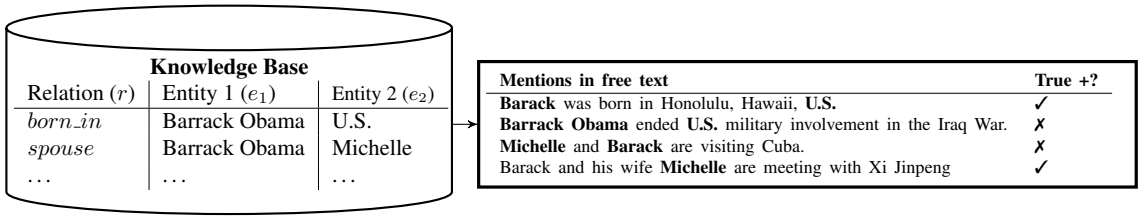

Figure 1: Illustration of the distant supervision paradigm and errors

set.

When an existing KB is available, a much larger set of known facts can be used to bootstrap training data, a procedure known as distant supervision (DS). DS automatically labels its own training data by heuristically aligning facts from a KB with an unlabeled corpus. The KB, written as D, can be seen as a collection of relational tables $r(e_1, e_2)$, in which $r \in R$ (R is the set of relation labels), and $< e_1, e_2 >$ is a pair of entities that are known to have relation r. The corpus is written as C.

The intuition underlying DS is that any sentence in C which mentions the same pair of entities (e_1 and e_2) expresses a particular relationship $\hat{r}$ between them, which most likely corresponds to the known fact from the KB, $\hat{r}(e_1, e_2) = r(e_1, e_2)$, and thus forms a positive training example for an extractor of relation r. DS has been successfully applied in many relation extraction tasks [23, 24] as it allows for the creation of large training sets with little or no human effort.

Equally apparent from the above intuition is the danger of finding incorrect examples for the intended relation. The heuristic of accepting each co-occurrence of the entity pair $< e_1, e_2 >$ as a positive training item because of the KB entry $r(e_1, e_2)$ is known to generate noisy training data or false positives [25], i.e., two entities co-occurring in text are not guaranteed to express the same relation as the field in the KB they were generated from. The same goes for the generation of negative examples: training data consisting of facts missing from the KB are not guaranteed to be false since a KB in practice is highly incomplete. An illustration of DS generating noisy training data is shown in Figure 1.

Several strategies have been proposed to reduce this noise. The most prominent make use of latent variable models, in which the assumption is made that each known fact is expressed at least once in the corpus [25, 26, 27]. These methods are cumbersome to train and are sensitive to initialization parameters of the model.

An active research direction is the combination of DS with partial supervision. Several recent works differ in the way this supervision is chosen and included. Some focus on active learning, selecting training

instances to be labeled according to an uncertainty criterion [28, 23], while others focus on annotations of surface patterns and define rules or guidelines in a semi-supervised learning setting [29]. Existing methods for fusion of distant and partial supervision require thousands of annotations and hours of manual labor for minor improvements (4% in F_1 for 23,425 annotations [28] or 2,500 labeled sentences indicating true positives for a 3.9% gain in F_1 [29]). In this work we start from a distantly supervised training set and demonstrate how noise can be reduced, requiring only 5 minutes of annotations per relation, while obtaining significant improvements in precision and recall of the extracted relations.

We define the following research questions:

RQ 1. How can we add supervision most effectively to reduce noise and optimize relation extractors?

RQ 2. Can we combine semi-supervised learning and dimension reduction techniques to further enhance the quality of the training data and obtain state-of-the-art results using minimal manual supervision?

With the following contributions, we provide answers to these research questions:

1. In answer to RQ 1, we demonstrate the effectiveness and efficiency of filtering training data based on high-precision trigger patterns. These are obtained by training initial weak classifiers and manually labeling a small amount of features chosen according to an active learning criterion.

2. We tackle RQ 2 by proposing a semi-supervised learning technique that allows extending an initial set of high-quality training instances with weakly supervised candidate training items by measuring their similarity in a low-dimensional semantic vector space. This technique is called Semantic Label Propagation.

3. We evaluate our methodology on test data from the English Slot Filling (ESF) task of the knowledge base population track at the 2014 Text Analysis Conference (TAC). We compare different methods by using them in an existing KBP system. Our relation extractors attain state-of-the-art effectiveness (a micro averaged F_1 value of 36%) while depending on a very low manual annotation effort (i.e., 5 minutes per relation).

In Section 2 we give an overview of existing supervised and semi-supervised RE methods and highlight their remaining shortcomings. Section 3 describes our proposed methodology, with some details on the DS starting point (Section 3.1), the manual feature annotation approach (Section 3.2), and the introduction of the semantic label propagation method (Section 3.3). The experimental results are given in Section 4, followed by our conclusions in Section 5.

2. Related Work

The key idea of our proposed approach is to combine DS with a minimal amount of supervision, i.e., requiring as few (feature) annotations as possible. Thus, our work is to be framed in the context of supervised and semi-supervised relation extraction (RE), and is related to approaches designed to minimize the annotation cost, e.g., active learning. Furthermore, we use compact vector representations carrying semantics, i.e., so-called word embeddings. Below, we therefore briefly summarize related work in the areas of (i) supervised RE, (ii) semi-supervised RE, (iii) evaluations of RE, (iv) active learning and (v) word embeddings.

2.1. Supervised Relation Extraction

Supervised RE methods rely on training data in the form of sentences tagged with a label indicating the presence or absence of the considered relation. There are three broad classes of supervised RE: (i) methods based on manual feature engineering, (ii) kernel based methods, and (iii) convolutional neural nets.

Methods based on feature-engineering [17, 30] extract a rich list of manually designed structural, lexical, syntactic and semantic features to represent the given relation mentions as sparse vectors. These features are cues for the decision whether the relation is present or not. Afterwards a classifier is trained on positive and negative examples. In contrast, *kernel based methods* [31, 32, 19] represent each relation mention as an object such as an augmented token sequence or a parse tree, and use a carefully designed kernel function, e.g., subsequence kernel or a convolution tree kernel, to calculate their similarity with test patterns. These objects are usually augmented with extra features such as semantic information. With the recent success of deep neural networks in natural language processing, Convolutional neural networks (CNNs) have emerged as effective relation extractors [33, 34, 35]. CNNs avoid the need for preprocessing and manual feature design by transforming tokens into dense vectors using embeddings of words and extract n-gram based features independent of the position in the sentence.

Supervised approaches all share the need for training data, which is expensive to obtain. Two common methods have emerged for the generation of large quantities of training data, both require an initial set of known instances. When this number is initially small, the technique of *bootstrapping* is used. When a very large number of instances is available from an existing knowledge base, *distant supervision* is the preferred technique. Both are briefly discussed below.

2.1.1. Bootstrapping models for Relation Extraction

When a limited set of labeled instances is available, bootstrapping methods have proven to be effective methods to generate high-precision relation patterns [21, 22, 36, 37]. The objective of bootstrapping is to expand an initial 'seed' set of instances with new relationship instances. Documents are scanned for entities from the seed instances and linguistic patterns connecting them are extracted. Patterns are then ranked according to coverage (recall) and low error rate (precision). Using the top scoring patterns, new seed instances are extracted and the cycle is repeated.

An important step in bootstrapping methods is the calculation of similarity between new patterns and the ones in the seed set. This measure decides whether a new pattern is relation oriented or not, based on the existing set. Systems use measures based on exact matches [36], cosine-similarity [21] or kernels [37]. A fundamental problem of these methods is semantic drift [38, 39]: bootstrapping, after several iterations, deviates from the semantics of the seed relationship and extracts unrelated instances which in turn generate faulty patterns. This phenomenon worsens with the number of iterations of the bootstrapping process.

Recently, Batista et al. [40] proposed the use of word embeddings for capturing semantic similarity between patterns. Contexts are modeled using linear combinations of the word embeddings and similarity is measured in the resulting vector space. This approach has shown to reduce semantic drift compared to previous similarity measures.

2.1.2. Distant Supervision

Distant supervision (DS) was first proposed in [41], where labeled data was generated by aligning instances from the Yeast Protein Database into research articles to train an extractor. This approach was later applied for training of relation extractors between entities [13] and jointly training the named entity classifier and the relation extractor [42].

Automatically gathering training data with DS is governed by the assumption that *all sentences* containing both entities engaged in a reference instance of a particular relation, represent that relation. Many methods have been proposed to reduce the noise in training sets from DS. In a series of works the labels of DS data are seen as latent variables. Riedel et al. [25] relaxed the strong *all sentences*-assumption to an *at-least-one-sentence*-assumption, creating a multi-instance learner. Hoffman et al. [43] modified this model by allowing entity pairs to express multiple relations, resulting in a multi-instance multi-label setting (MIML-RE). Surdeanu et al. [27] further extended this approach and included a secondary classifier, which jointly modeled all the sentences in texts and all labels in knowledge bases for a given entity pair.

Other methods apply heuristics [44], model the training data as a generative process [45, 46] or use a low-rank representation of the feature-label matrix to exploit the underlying semantic correlated information.

2.2. Semi-supervised Relation Extraction

Semi-supervised Learning is situated between supervised and unsupervised learning. In addition to unlabeled data, algorithms are provided with some supervised information. The training data comprises labeled instances $X_l = (x_1 \dots x_l)$ for which labels $Y_l = (y_1 \dots y_l)$ are provided, and typically a large set of unlabeled ones $X_u = (x_1 \dots x_u)$.

Semi-supervised techniques have been applied to RE on multiple occasions. Chen et al. [47] apply label propagation by representing labeled and unlabeled examples as nodes and their similarities as the weights of edges in a graph. In the classification process, the labels of unlabeled examples are then propagated from the labeled to unlabeled instances according to similarity. Experimental results demonstrate that this graph-based algorithm can outperform SVM in terms of F_1 when very few labeled examples are available. Sun et al. [18] show that several different word cluster-based features trained on large corpora can compensate for the sparsity of lexical features and thus improve the RE effectiveness.

Zhang et al. [48] compare DS and complete supervision as training resources but do not attempt to fuse them. They observe that DS systems are often recall gated: to improve DS quality, large input collections are needed. They also report modest improvements by adding crowd-sourced yes/no votes to the training instances. Training instances were selected at random as labeling using active learning criteria did not affect performance significantly.

Angeli et al. [28] show that providing a relatively small number of mention-level annotations can improve the accuracy of MIML-RE. They introduce an active learning criterion for the selection of instances incorporating both the uncertainty and the representativeness, and show that the choice of criterion is important. The MIML-RE model of Surdeanu et al. [27] marginally outperforms the Mintz++ baseline using solely DS: initialization of the latent variables using labeled data is needed for larger improvements. For this, a total of $10,000$ instances were labeled, resulting in a 3% increase on the micro-F_1 .

Guided DS as proposed by Pershina et al. [29] incorporates labeled patterns and trigger words to guide MIML-RE during training. They make use of a labeled dataset from TAC KBP to extract training guidelines, which are intended to generalize across many examples.

2.3. TAC KBP English Slot Filling

The knowledge base population (KBP) shared task is part of the NIST Text Analysis Conference and aims to evaluate different approaches for discovering facts about entities and expansion of knowledge bases. A selection of entities is distributed among participants for which missing facts need to be extracted from a given large collection of news articles and internet fora. Important components of these systems are query expansion, entity linking and relation extractors. Over the years DS has become a regular feature of effective systems [23, 49]. Other approaches use hand-coded rules or are based on question answering systems [49]. The top performing 2014 KBP ESF system [50] uses DS, the manual labeling of $100,000$ features, and is built on DeepDive, a database system allowing users to rapidly construct sophisticated end-to-end knowledge base population techniques [51]. After initial DS, features are manually labeled and only pairs associated with labeled features are used as positive examples. This approach has proven to be very effective but further investigation is needed to reduce the amount of feature labeling. Here, we show how we can strongly reduce this effort while maintaining high precision.

2.4. Active Learning and Feature Labeling

Active learning is used to reduce the amount of supervision required for effective learning. The most popular form of active learning is based on iteratively requiring manual labels for the most informative instances, an approach called uncertainty sampling. In relation extraction, typical approaches include query-by-committee [28, 52] and cluster-based sampling [53]. While the focus in RE has been on labeling relation instances, alternative methods have been proposed in other tasks in which features (e.g., patterns, or the occurrence of terms) are labeled as opposed to instances [54, 55], resulting in a higher performance using less supervision.

Getting positive examples for certain relations can be hard, especially when training data is weakly supervised. Standard uncertainty sampling is ineffective in this case: it is likely that a feature or instance has a low certainty score because it does not carry much discriminative information about the classes. Assigning labels to the most certain features has much greater impact on the classifier and can remove the principle sources of noise. This approach has been coined as feature certainty [55], and we show that this approach is especially effective in DS for features that generalize across many training instances.

2.5. Distributional Semantics

The Distributional Hypothesis [56] states that words that tend to occur in similar contexts are likely to have similar meanings. Representations of words as dense, low-dimensional vectors (as opposed to the stan-

dard one-hot vectors), called word embeddings, exploit this hypothesis and are trained from large amounts of unlabeled text. Representations for words will be similar to those of related words, allowing the model to generalize better to unseen events. The resulting vector space is also called a *vector model of meaning* [57]. Common techniques for generating very dense, short vectors use dimensionality reduction techniques (e.g., singular value decomposition) or neural nets to create so-called word embeddings. Word embeddings have proven to be beneficial for many natural language processing tasks including POS-tagging, machine translation and semantic role labeling. Two prominent methods for the embedding of words are *Word2Vec* [58] and *GloVe* [59].

While much research has been directed at ways of constructing distributional representations of individual words, for example co-occurrence based representations and word embeddings, there has been far less consensus regarding the representation of larger constructions such as phrases and sentences from these representations. Blacoe et al. [60] show that, for short phrases, a simple composition like addition or multiplication of the distributional word representations is competitive with more complex supervised models such as recursive neural networks.

3. Labeling Strategy for Noise Reduction

In this section we introduce our strategy to combine distantly supervised training data with minimal amounts of supervision. Briefly summarized, we designed our labeling strategy such as to *minimize the amount of false positive instances or noise while maintaining the diversity of relation expressions generated by DS.*

We perform a highly selective form of noise reduction starting from a fully distantly supervised relation extractor, described in Section 3.1, and use the feature weights of this initial extractor to guide manual supervision in the feature space. Various questions arise from this. When do we over-constrain the original training set generated by DS? What is the trade-of between the application of DS with highly diverse labeled instances, and the constraining approach of labeling features, with a highly accurate yet restricted set of training data? This is discussed in detail in Sections 3.2 and 3.3.

Our approach is depicted in Figure 2, and comprises the following steps:

(1) An existing KB is used to generate distantly supervised training instances by matching its facts with sentences from a large text corpus. We discuss the characteristics of this weakly labeled training set as well as the features extracted from each sentence (see Section 3.1).

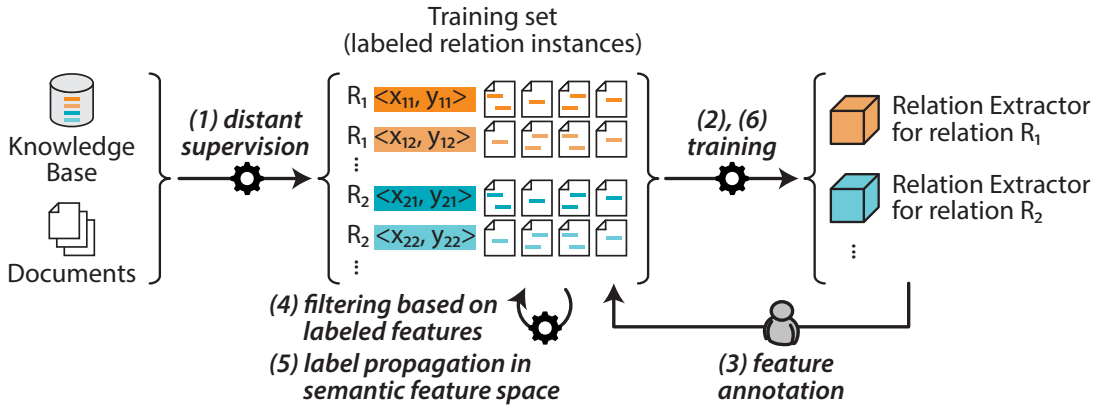

Figure 2: Workflow Overview. Note that only step (3) involves human annotations.

(2) An initial relation extractor is trained using the noisy training data generated in step (1).

(3) Confident positive features learned by this initial classifier are presented to an annotator with knowledge of the semantics of the relation and labeled as true positive or false positive.

(4) The collection of training instances is filtered according to the labeled features and a second classifier is trained. This framework, in which we combine supervision and DS, is explained in Section 3.2.

(5) In a *semi-supervised* step, the filtered distantly supervised training data is added to training data by propagating labels from labeled features to distantly supervised instances based on similarity in a semantic vector space of reduced dimension. The technique is presented in Section 3.3 as *Semantic Label Propagation*.

(6) A final relation extractor is trained on the augmented training set. We evaluate and discuss results of the proposed techniques in Section 4.

3.1. Distantly Supervised Training Data

The English Gigaword corpus [61] is used as unlabeled text collection to generate relation mentions. The corpus consists of 1.8 million news articles published between January 1987 and June 2007. Articles are first preprocessed using different components of the Stanford CoreNLP toolkit [62], including sentence segmentation, tokenizing, POS-tagging, named entity recognition, and clustering of noun phrases which refer to the same entity.

As KB we use a snapshot of Freebase (now Wikidata) from May 2013. The relation schema of Freebase is mapped to that used for evaluation, the NIST TAC KBP ESF Task, which defines 41 relations, including 25 relations with a person as subject entity and 16 with organizations as subject. 26 relations require objects or fillers that are themselves named entities (e.g., Scranton as place of birth of Joe Biden), whereas others require string-values (e.g., profession (senator, teacher,...), cause of death (cancer, car accident,...)).

We perform weak entity linking between Freebase entities and textual mentions using simple surface string matching. We reduce the effect of faulty entity links by thresholding the amount of training data per subject entity [63]. Most frequently occurring entities from the training data (e.g., John Smith, Robert Johnson, ...) are often most ambiguous, hard to link to a KB and thus result in noisy training data. Thresholding the amount of training data per entity also prevents the classifier from overfitting on several, popular entities. This follows from the observation that training data is initially skewed towards several entities frequently occurring in news articles, like Barack Obama or the United Nations, resulting in over-classifying professions of persons as president or seeing countries as members of the organization.

For each generated pair of mentions, we compute various lexical, syntactic and semantic features. Table 1 shows an overview of all the features applied for the relation classification. We use these features in a binary logistic regression classifier. Features are illustrated for an example relation-instance <Ray Young, General Motors> and the sentence *"Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi"*.

For each relation R_i, we generate a set of (noisy) positive examples denoted as R_i^+ and defined as

$$R_i^+ = \{ (m_1, m_2) \mid R_i(e_1, e_2) \wedge EL(e_1, m_1) \wedge EL(e_2, m_2) \}$$

with e_1 and e_2 being subject and object entities from the KB and $EL(e_1, m_1)$ being the entity e_1 linked to mention m_1 in the text. As in previous work [43, 30], we impose the constraint that both entity mentions $(m_1, m_2) \in R_i^+$ are contained in the same sentence. To generate negative examples for each relation, we sample instances from co-occurring entities for which the relation is not present in the KB.

We measured the amount of noise, i.e., false positives, in the training set of positive DS instances, for a selection of 15 relations: we manually verified 2,000 randomly chosen instances (that DS found as supposedly positive examples) for each of these relations. Table 2 shows the percentage of true positives among these 200 instances for each of these relations, which strongly varies among relations, ranging from 10% to 90%.

Table 1: Overview of different features used for classification for the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi".

Feature	Description	Example Feature Value
Dependency tree	Shortest path connecting the two names in the dependency parsing tree coupled with entity types of the two names	PERSON←appos←officer →prep_of→ ORGANIZATION
	The head word for name one	said
	The head word for name two	officer
	Whether *1dh* is the same as *e2dh*	false
	The dependent word for name one	officer
	The dependent word for name two	nil
Token sequence features	The middle token sequence pattern	, the chief financial officer of
	Number of tokens between the two names	6
	First token in between	,
	Last token in between	of
	Other tokens in between	{the, chief, financial, officer}
	First token before the first name	nil
	Second token before the first name	nil
	First token after the second name	,
	Second token after the second name	said
Entity features	String of name one	Ray_Young
	String of name two	General_Motors
	Conjunction of *e1* and *e2*	Ray_Young–General_Motors
	Entity type of name one	PERSON
	Entity type of name two	ORGANIZATION
	Conjunction of *et1* and *et2*	PERSON–ORGANIZATION
Semantic feature	Title in between	True
Order feature	1 if name one comes before name two; 2 otherwise.	1
Parse Tree	POS-tags on the path connecting the two names	NNP→DT→JJ→JJ →NN→IN→NNP

Table 2: Training Data. Fractions of true positives are estimated from the training data by manually labeling a sample of 2,000 instances per relation that DS indicated as positive examples

.

Relation	**Estimated Fraction of True Positives**	**Positively Labeled SDPs**	**Remaining Training Data after Filtering**	**Initial Number of True Positives**
per:title	85.1%	157	26.2%	369,079
org:top_members_employees	71.7%	236	16.7%	93,900
per:employee_or_member_of	87.8%	256	16.5%	260,785
per:age	62.4%	79	52.2%	58,980
per:origin	85.2%	116	11.9%	1,555,478
per:countries_of_residence	55.6%	65	8.4%	493,064
per:charges	59.4%	122	21.5%	17,639
per:cities_of_residence	11.7%	96	7.4%	370,153
per:cause_of_death	51.9%	97	29.4%	31,386
per:spouse	63.2%	124	12.1%	172,874
per:city_of_death	19.9%	92	5.6%	125,333
org:country_of_headquarters	10.8%	92	13.4%	13,435
per:country_of_death	77.6%	70	16.5%	128,773
org:city_of_headquarters	56.5%	67	42.7%	36,238
org:founded_by	13.3%	85	22.7%	318,991

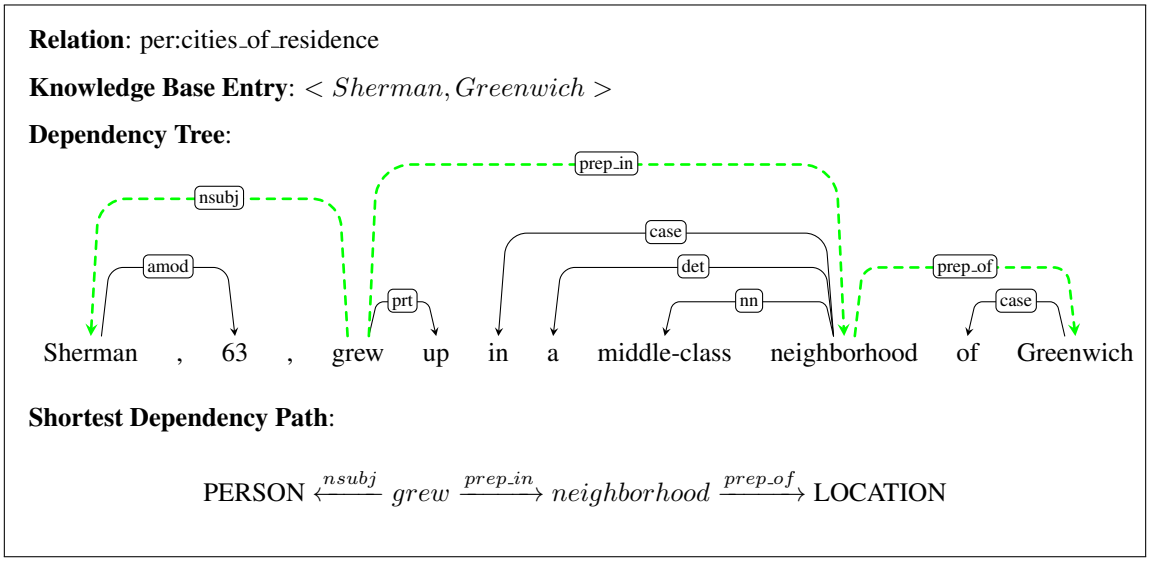

Figure 3: Dependency tree feature

3.2. Labeling High Confidence Shortest Dependency Paths

This section describes the manual feature labeling step that allows transforming a full DS training set into a strongly reduced yet highly accurate training set, based on feature labeling. We focus on a particular kind of feature, i.e., a relation's shortest dependency path (SDP). Dependency paths have empirically been proven to be very informative for relation extraction: their capability of capturing information is evidenced by a systematic comparison in effectiveness of different kernel methods [64] or as features in feature-based systems [17]. This was originally proposed by Bunescu et al. [19], who claimed that the relation expressed by a sentence is often captured in the *shortest* path connecting the entities in the dependency graph. Figure 3 shows an example of an SDP for a sentence expressing a relation between a person and a city of residence.

As shown in Table 2, the fraction of false positive items among all weakly supervised instances can be very large. Labeling features based on the standard active learning approach of uncertainty sampling is ineffective in our case since it is likely that a feature or instance has a low certainty score simply because not much discriminative information about the classes is carried. Annotating many such instances would be a waste of effort. Assigning labels to the most certain features has much greater impact on the classifier and can remove the principal sources of noise. This approach is called feature certainty sampling [55]. It is intuitively an attractive method, as the goal is to reduce the most influential sources of noise as quickly as

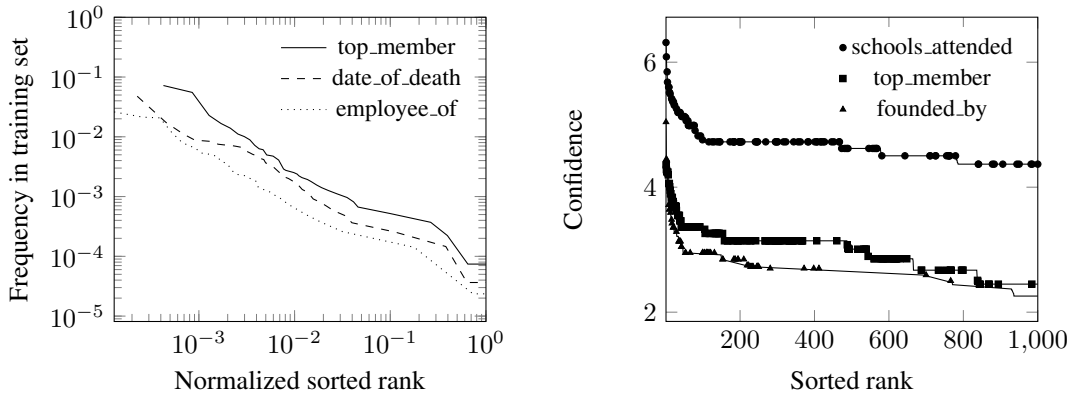

Figure 4: Illustration of frequency and confidence of dependency paths for example relations. (a) Occurrence frequency, ranked from highest to lowest, and (b) confidence C of dependency paths (eq. 1), ranked from highest to lowest, with indication of true positives.

possible. For example for the relation *founded_by*, there are many persons that founded a company who were also *top_members*, leading to instances that we wish to remove when cleaning up the training data for the relation *founded_by*. SDPs offer all the information needed to assess the relationship validity of the training instances, are easily labeled, and generalize over a considerable fraction of the training set as opposed to many of the feature-unigrams which remain ambiguous in many cases. We implement the feature certainty idea by ranking SDP features according to the odds that when a particular SDP occurs, it corresponds to a valid relation instance. This corresponds to ranking by the following quantity, which we call the considered SDP's confidence

$$\text{Confidence}(SDP) = \frac{P(+|SDP)}{P(-|SDP)}. \tag{1}$$

It can be directly estimated from the original DS training set, based on each SDP feature's (smoothed) occurrence frequencies among the positive and negative distantly supervised instances. In particular, $P(+|SDP)$ indicates the SPD's fraction of occurrences among the positive training data and $P(-|SDP)$) among the negative.

All dependency paths are ranked from most to least confident and the top-k are assigned to a human annotator to select the true positive SDPs. The annotator is asked to select only the patterns which unambiguously express the relation. That is, a pattern is accepted only if the annotator judges it a sufficient condition for that relation. The annotator is provided with several complete sentences containing the dependency path to this cause. When the SDP does not include any verbs, e.g., when entities are both part of the

same Noun Phrase like "Microsoft CEO Bill Gates", all words between the subject and object are included and the complete path is added to the filter set. In our experiments, we restrict the time of SDP annotations to a limited effort of 5 minutes for each relation. On average our expert annotator was able to label around 250 SDPs per relation this way. The ease of annotating SDPs becomes apparent when compared with annotating random relation instances, of which they managed to annotate only 100 in the same period of time. Section 4.3 provides further details on the different annotation methodologies for the experiments.

The motivation behind limiting the annotation time per relation to only a few hundred patterns comes from the following analysis. First of all, a small subset of all different patterns is responsible for the majority of relation instances in the DS training set. In fact, the sparsity of distantly supervised training data becomes apparent when extracting all SDPs for each fact in the KB in one pass over the corpus. Figure 4a shows the approximately zipfian distribution of the frequency of the dependency paths generated by DS in the positively labeled training set for several example relations. The abscis shows the rank of dependency paths for various relations, sorted from most to least frequent, normalized by the total number of paths for the respective relations (to allow visualization on the same graph). In line with our goal of getting a highly accurate training set with the largest sources of noise removed at a low annotation cost, we focused on capturing those top most frequent patterns. Secondly, we noticed that beyond the first few hundred most confident SDPs, which took around 5 minutes to annotate, further true positives tend to occur less frequently. Annotating many more SDPs would only marginally increase the diversity in the training set, at a rapidly increasing annotation cost. Figure 4b illustrates the occurrence of true positive patterns for decreasing confidence scores. For several example relations, the figure shows the true positive patterns as markers on the confidence distribution of the $1,000$ most confident SDPs.

Finally, using the manually selected set of SDPs, the complete training set is filtered by enforcing that one of these SDPs be present in the feature set of the instance. We include all mention pairs associated with that feature as positive examples of the considered relation. The classifier trained on the resulting training set is intuitively of high precision but doesn't generalize well to unseen phrase constructions. Note that the classifier is quite different from a regular pattern based relation extractor. Although all training instances satisfy at least one of the accepted SDPs, the classifier itself is trained on a set of features including, but not restricted to, these SDPs (see Table 1). Still, most of the benefits of DS are lost by having the selection of training instances governed by a limited set of patterns.

The fourth column of Table 2 lists the fraction of training data remaining after filtering out all patterns apart from those classified as indicative of the relation at hand. The amount of training data remaining after

Table 3: Examples of top-ranked patterns

Relation	**Top SDP**	**Assessment**
top_members_employees	PER $\xleftarrow{appos}$ *executive* $\xrightarrow{prep\_of}$ ORG	✓
	PER $\xleftarrow{appos}$ *chairman* $\xrightarrow{appos}$ ORG	✓
	ORG $\xleftarrow{nn}$ *founder* $\xrightarrow{prep\_of}$ PER	✗
children	PER-2 $\xleftarrow{appos}$ *son* $\xrightarrow{prep\_of}$ PER-1	✓
	PER-1 $\xleftarrow{appos}$ *father* $\xrightarrow{prep\_of}$ PER-2	✓
	PER-2 $\xleftarrow{nn}$ *grandson* $\xrightarrow{prep\_of}$ PER-1	✗
city_of_birth	PER $\xleftarrow{rcmod}$ *born* $\xrightarrow{prep\_in}$ LOC	✓
	PER $\xleftarrow{nsubj}$ *mayor* $\xrightarrow{prep\_of}$ LOC	✗
	PER $\xleftarrow{appos}$ *historian* $\xrightarrow{prep\_from}$ LOC	✗
schools_attended	PER $\xleftarrow{nsubj}$ *graduated* $\xrightarrow{prep\_from}$ ORG	✓
	PER $\xleftarrow{dep}$ *student* $\xrightarrow{prep\_at}$ ORG	✓
	PER $\xleftarrow{appos}$ *teacher* $\xrightarrow{prep\_at}$ ORG	✗
(org:)parents	ORG-2 $\xleftarrow{appos}$ *subsidiary* $\xrightarrow{prep\_of}$ ORG-1	✓
	ORG-1 $\xleftarrow{appos}$ *division* $\xrightarrow{prep\_of}$ ORG-2	✓
	ORG-2 $\xleftarrow{prep\_to}$ *shareholder* $\xrightarrow{dep}$ ORG-1	✗

this filtering step strongly depends on the specific relation, varying from 5% to more than half of the original training set. Yet on the whole, the filtering results in a strong reduction of the purely DS-based training data, often removing much more than the actual fraction of noise (column 2). For example, for the relation *per:employee_or_member_of*, we note only $100\% - 87.8\% = 12.2\%$ false positives, but the manual filtering leads to discarding 83.5% of the DS instances.

The strategy described in the previous paragraphs is related to the *guidelines* strategy from Pershina et al. [29] (without the MIML model) in labeling features, but it differs in some essential aspects. Instead of needing a fully annotated corpus to do so, we rank and label features entirely based on DS. Labeling features based on a fully labeled set ignores the variety of DS and risks being biased towards the smaller set of labeled instances. Also, no active learning criteria were applied when choosing which features to label, making the process even more efficient.

3.3. Noise Reduction using Semantic Label Propagation

If we strictly follow the approach proposed in Section 3.2 and only retain DS training instances that satisfy a positively labeled SDP, an important advantage of DS is lost, namely its potential of reaching high recall. If we limit the feature annotation effort, we risk losing highly valuable SDPs. To counteract this effect, we introduce a second (re)labeling stage, adopting a semi-supervised learning (SSL) strategy to expand the training set. This is done by again adding some instances from the set of previously discarded DS instances with SDPs not matching any of the manually labeled patterns. We rely on the basic SSL approach of self-training by propagating labels from known instances to the nearest neighboring unlabeled instances. This method requires a method of determining the distance between labeled and unlabeled instances. Dangers of self-training include the failure to expand beyond the initial training data or the introduction of errors into the labeled data. In order to avoid an overly strong focus on the filtered training data, we use low-dimensional vector representations of words, also called word embeddings.

Word embeddings allow for a relaxed semantic matching between the labeled seed patterns and the remaining weakly labeled patterns. As shown by Sterckx et al. [53], representing small phrases by summing each individual word's embedding leads to semantic representations of small phrases that are meaningful for the goal of relation extraction. We represent each relation instance by a single vector by first removing stop-words and averaging the embeddings of the words on the dependency path. For example, consider the sentence:

Geagea on Friday for the first time addressed the court judging him for **murder** charges.

which has the following SDP,

$$\text{PER} \xleftarrow{nsubj} addressed \xrightarrow{dobj} court \xrightarrow{vmod} judging \xrightarrow{prep\_for} charges \xrightarrow{nn} \text{Criminal\_Charge}$$

Its low-dimensional representation $\vec{C}$ is hence generated as

$$\vec{C} = \frac{E(\text{“addressed”}) + E(\text{“court”}) + E(\text{“judging”}) + E(\text{“charges”})}{4}, \tag{2}$$

with $E(x)$ the word embedding of word x. The similarity between a labeled pattern $\vec{C}_t$ and a weakly labeled pattern $\vec{C}_{DS}$ is then measured using cosine similarity between the vector representations.

$$Sim(\vec{C}_t, \vec{C}_{DS}) = \frac{\vec{C}_t.\vec{C}_{DS}}{|\vec{C}_t|.|\vec{C}_{DS}|} \tag{3}$$

In the special case that no verbs occur between two entities, all the words between the two entities are used to build the representations for the context vector.

Using these low-dimensional continuous representations of patterns, we can calculate similarities between longer, less frequently occurring patterns in the training data and the patterns from the initial seed set which are the most frequently occurring ones. We can now increase recall by adding similar but less frequent patterns. More specifically, we calculate the similarity of the average vector of the labeled patterns (as in the Rocchio classifier type of self-training) with each of the remaining patterns in the DS set and extend the training data with the patterns that have a sufficiently high similarity with the labeled ones. We call this technique *Semantic Label Propagation*.

4. Experimental Results

4.1. Testing Methodology

We evaluate the relation extractors in the context of a Knowledge Base Population system [63, 65] using the NIST TAC KBP English Slot Filling (ESF) Evaluation from 2012 to 2014. We choose for this evaluation because of the diversity and difficulty of entities in the queries. In the end-to-end ESF framework, the input to the system is a given entity (the 'query'), a set of relations, and a collection of articles. The output is a set of slot fillers, where each slot filler is a triple consisting of two entities (including the query entity) and a relation predicted to hold among these entities.

4.2. Knowledge Base Population System

Systems participating in the TAC KBP ESF need to handle each task of filling missing slots in a KB. Participants are only provided with one surface-text occurrence of each query entity in a large collection of text provided by the organizers. This means that an information retrieval component is needed to provide the relation extractor with sentences containing candidate fillers. Our system performs query expansion using Freebase aliases and Wikipedia pages. Each document containing one of the aliases is parsed and named entities are automatically detected. Persons, organizations, and locations are recognized, and locations are further categorized as cities, states, or countries. Non-entity fillers like titles or charges are tagged using lists and table-lookups. For further details of the KBP system we refer to [63, 65].

4.3. Methodologies for Supervision

In this section we detail the different procedures for human supervision. Supervision is obtained in two forms: by labeling shortest dependency paths (SDPs) and by labeling single training instances indicated as positive by DS, as either true positives or as false positives (noise). After a background corpus is linked with a knowledge base, phrases containing facts are stored in a database for further feature extraction, post processing, and calculation of feature confidence values. Our annotators for the labeling of single training instances were undergraduate students from different backgrounds with little or no experience in machine learning or natural language processing. First, they were briefed on the semantics of the relation to be extracted using the official TAC KBP guidelines. They were then presented with training instances, i.e., phrases from the database. Each instance was shown with entity and subject highlighted and colored. The average time needed to annotate a batch of 2,000 instances was three hours, corresponding to about 5 seconds per instance, including the time needed to read and judge the sentence. As this procedure was relatively expensive (annotators were paid \$15 per hour), only the 15 most frequent relations, strongly influencing the optimal micro-F_1 score shown in Table 2, were selected. Other relations received between 200 and 1,000 annotations each. In contrast, the time for annotation of the SDPs was limited to merely 5 minutes per relation, during which, on average, 200 SDPs were judged. SDPs were presented in a spreadsheet as a list, and true positives were labeled using a simple checkbox. All SDP annotations were done by a single expert annotator. To measure the degree of expertise needed for these annotations, we also assigned a novice annotator (student) with the same task. We measured annotator agreement and time needed for a selection of the relations. For this experiment the student was explained the meaning of dependency paths and the aim of choosing valid SDPs. Several lists of SDPs that the expert was able to label in 5 minutes were presented to the student. For the first two relations the student needed more than 10 minutes to label, but for the subsequent relations, annotation time dropped to 5 minutes per relation, equivalent to the time needed by an expert annotator. We measured inter annotator agreement using Cohen's kappa coefficient κ. Inter-annotator agreement between student and expert was initially moderate ($\kappa = 0.65$) and increased after the student completed lists of SDPs for two relations (κ varies between 0.85 and 0.95), indicating a very good agreement.

4.4. Pattern-based Restriction vs. Similarity-based Extension

As Table 2 shows, applying the manually annotated features as described in Section 3.2 often leads to a drastic reduction of training instances, compared to the original distantly labeled training set. Using

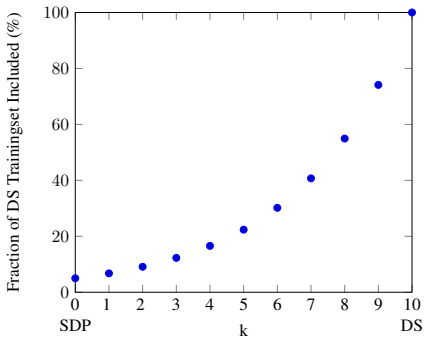

Figure 5: Example of the proposed sampling strategy for training set sizes, with $N_{filtered} = 0.05N_{DS}$, and in $K = 10$ steps.

similarity metrics described in Section 3.3, we again add weakly supervised training data to the filtered data. An important question is therefore how to optimally combine initial reduction with subsequent expanding of the training instances. Intuitively, one would expect a high-precision-low-recall effect in the extreme case of adding no similar patterns, and a low-precision-high-recall effect when adding all weakly labeled patterns, both leading to a sub-optimal F_1 measure. On the other hand, adding a limited amount of similar patterns may increase recall without harming precision too much. In this section, we investigate for a selection of relations, how the quality of the training set depends on the fraction of similar patterns it is extended with. In our experimental setup, we start from the training set that only contains the $N_{filtered}$ instances that match the manually labeled patterns, gradually adding weakly labeled data, and each time training binary classifiers on the corresponding training set. We chose to let the additional data grow exponentially, which allows studying the effect of adding few extra instances initially, but extending towards the full weakly supervised training set of size N_{DS} in a limited number of cases. More specifically, in K experiments of adding additional instances, the intermediate training set size N_k at step k is given by

$$N_k = N_{filtered} \cdot \left(\frac{N_{DS}}{N_{filtered}} \right)^{k/K} \tag{4}$$

Figure 5 illustrates how an initial training set containing only 5% of the amount of instances from the full weakly labeled training set, is increased in $K = 10$ consecutive experiments.

Apart from studying the addition of varying amounts of similar patterns, in this section we also investigate the influence of the type of similarity measure used. In Section 3.2 we suggested the use of word embeddings, but is there a difference between different types of embeddings? Would embeddings work better than traditional dimension reduction techniques? And would such techniques indeed perform better

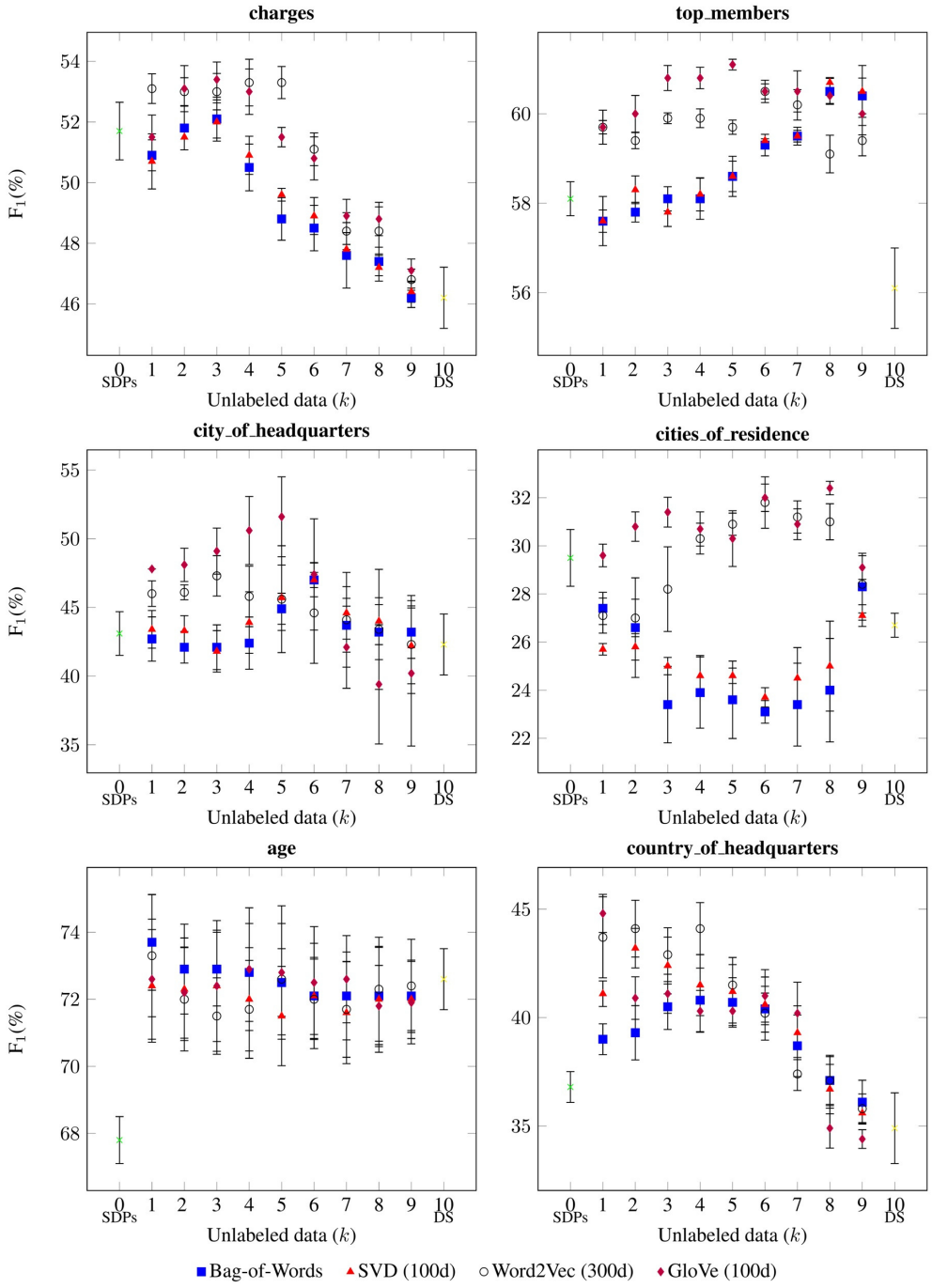

Figure 6: Illustration of the behavior of Semantic Label Propagation for different dimension reduction techniques, and different amounts of added weakly labeled data, quantified by k (as in eq. 4), with $K = 10$. $k = 0$ corresponds to only accepting manually filtered SDPs, and $k = 10$ corresponds to using all weakly labeled (DS) data for training.

than the original one-hot vector representations? These questions can be answered by considering several similarity measures. As a classical baseline, we represent SDPs using the average one-hot or bag-of-words (BOW) representations of the words contained in the SDPs. We also transform the set of one-hot representations using singular value decomposition (SVD) [66] fitted on the complete training set. For representations using the summed average of word embeddings described in Section 3.3, we use two sets of pre-trained *Word2Vec* embeddings[1] (trained on news text) and *GloVe* embeddings[2] (trained on Wikipedia text).

Figure 6 shows the effect of adding different amounts of weakly labeled data, for different values of k as in eq. 4 (with $K = 10$ steps) and for similarity measures based on the different types of representations described above. Six frequently occurring relations were selected such that they give an idea of the various forms of behavior that we observed during our investigation of all extracted relations. The chosen effectiveness measure is the optimal F_1 value of classification on a development set, consisting of training data from 2012 and 2013. (In the next Section we will evaluate on a held-out test set, which consists of queries from the 2014 TAC ESF task, whereby the optimal value of k and type of dimension reduction is selected based on the development set.) Also shown are standard deviations on these optimal F_1 -values, obtained by resampling different positive and negative instances for training the classifier. Several insights can be gained from Fig. 6:

- *SDPs vs full DS training set:* We observe that the effect of expanding the initial training set is strongly dependent on the specific relation and the quality of the initial training data. In many cases training data filtered using only highly confident SDPs ($k = 0$) generates a better relation extractor than pure DS ($k = K$). This holds for all shown relations, except for the *age* relation. We have to be aware that wrongly annotating an important pattern, or by chance missing any in the top most confident ones, can strongly reduce recall when only using the accepted SDPs. Adding even a small amount of similar patterns may hence result in a steep increase in effectiveness, such as for $k = 1$ in the *age* and *country_of_headquarters* relations.
- *Effect of semantic label propagation:* When relaxing the filtering (i.e., increasing k) by adding unlabeled data, the optimal F_1 tends to increase until a certain point, and then again drops towards the behavior of a fully DS training set, because the quality or similarity of the added training data declines and too many false positives are re-introduced. The threshold on the amount of added DS instances is thus an important

[1] `https://code.google.com/p/word2vec/`

[2] `http://nlp.stanford.edu/projects/glove/`

parameter to tune on a development set. For some of the relations there is an optimal amount of added unlabeled data, whereas other relations show no clear optimum and fluctuate between distant and filtered classifiers' values.

- *Impact of dimensionality reduction:* The use of word embeddings often leads to an improved maximum F_1 value with respect to the BOW-representations or SVD-based dimension reduction. This is for example very clear for the *charges*, *city_of_headquarters*, or *cities_of_residence* relations, with a slight preference of the *GloVe* embeddings with respect to *Word2Vec* for this application. However, we also noticed that word embeddings are not always better than the BOW or SVD based representations. For example, the highest optimal F_1 for the *age* relation is reached with the BOW model.

4.5. End-to-End Knowledge Base Population Results

This section presents the results of training binary relation classifiers according to our new strategy for each of the 41 relations of the TAC KBP schema. We tuned hyperparameters on data of the 2012 and 2013 tracks and now test on the last edition of the ESF track of 2014.

Next to the thresholds of choosing the amount of unlabeled data added as discussed previously (i.e., the value of k), other parameters include regularization and the ratio between positive and negative instances, which appeared to be an important parameter influencing the confidence of an optimal F_1 value greatly. Different ratios of negative to positive instances resulted in shifting the optimal trade-off between precision and recall. The amount of available negative training data was on many occasions larger than the available positive. More negative than positive training data overall appeared to result in lower positive classification probabilities assigned by the classifier to test instances. Negative instances had to be down-weighted multiple times to prevent the classifier from being too strict and rarely classify a relation as true. For each relation, this parameter was tuned for optimal F_1 value at the 0.5 probability threshold of the logistic regression classifier.

We use the official TAC KBP evaluation script which calculates the micro-average of all classifications. All methods are evaluated while ignoring provenances (the character offsets in the documents which contain the justification for extraction of the relation), so as not to penalize any system for finding a new provenance not validated in the official evaluation key. A listing of precision, recall and F_1 for the top 20 most frequently occurring relations in the test set is shown in Table 4.

Next to traditional distant supervision (also known as *Mintz++*[30], indicated as 'distant Supervision' in Table 4), we compare our new semi-supervised approach ('Semantic Label Propagation') to a fully supervised classifier trained by manually labeling $50,000$ instances ('Fully Supervised'), and to the classifiers

obtained by purely filtering on manually labeled patterns ('SDP Filtered'). We also use the fully supervised classifiers in a traditional self-training scheme, classifying distantly supervised instances in the complete feature space and adding confident instances to the training set ('Self-Training (Instances)'). The supervision needed for these classifiers required far more annotation effort than the feature certainty sampling of Semantic Label Propagation.

The official F_1 value of 36.4% attained using Semantic Label Propagation is equivalent to the second best entry out of eighteen submissions to the 2014 ESF track [23]. A relation extractor is but a part of a KBP system and is influenced by each of the other modules (e.g., recognition and disambiguation of named entities), which makes it hard to compare to other systems. This is the case for the absolute values of Table 4, but still, it demonstrates the overall quality of our relation extractors. Especially, our system relying on very limited annotations has a competitive place among systems that rely on many hours of manual feature engineering [50]. Comparing the results for Semantic Label Propagation with the other approaches shows that the proposed method that combines a small labeling effort based on feature certainty with the Semantic Label Propagation technique, outperforms the DS method, semi-supervision using instance labeling, and full supervision methods. This is also confirmed in Fig. 7, which shows the trade-off between the precision and recall averaged over all TAC KBP relations for the different methods described above, using the TAC KBP evaluation script (varying the thresholds on classification).

Table 4: Results for Frequent Relations and official TAC-scorer

	Distant Supervision (Mintz++)			**SDP Filtered**			**Fully Supervised**			**Self-Training (Instances)**			**Semantic Label Propagation**		
Relation	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
title	22.3	58.8	32.3	36.1	39.1	37.5	28.0	61.1	38.4	36.5	43.2	**39.6**	37.3	41.2	39.2
top_members_employees	50.6	63.4	56.3	51.3	63.4	56.7	62.6	53.9	57.9	56.3	63.4	59.6	63.5	62.5	**63.0**
employee_or_member_of	31.4	34.0	32.6	33.8	51.0	**40.7**	23.5	45.7	31.0	32.2	40.4	35.8	27.9	51.0	36.1
age	71.6	72.5	72.0	75.6	70.0	72.7	68.0	62.5	64.9	73.6	70.0	71.8	68.8	82.5	**75.0**
origin	100.0	23.0	37.4	28.5	80.0	42.0	29.4	66.6	40.8	27.5	73.3	40.0	31.7	86.6	**46.4**
countries_of_residence	100.0	23.0	37.4	22.4	84.6	35.4	22.2	92.3	35.8	50.0	38.4	**43.4**	35.2	46.1	39.9
charges	45.0	52.9	48.6	40.9	52.9	46.1	70.4	44.1	**54.2**	47.6	58.8	52.6	44.3	68.1	53.7
cities_of_residence	22.9	45.8	30.5	31.5	25.0	27.9	11.2	62.5	19.0	36.3	16.6	22.8	34.4	41.6	**37.7**
cause_of_death	30.7	36.3	33.3	29.4	45.4	35.7	28.3	31.8	29.9	37.5	27.2	31.5	33.3	45.4	**38.4**
spouse	50.0	45.4	47.6	50.0	45.4	47.6	75.0	27.2	39.9	35.7	45.4	40.0	71.4	45.4	**55.5**
city_of_death	100.0	16.6	28.5	14.2	16.6	15.3	5.2	100.0	9.9	20.0	16.6	18.1	20.0	33.3	**25.0**
country_of_headquarters	22.7	41.6	29.4	62.5	41.6	**50.0**	25.0	50.0	33.3	100.0	25.0	40.0	100.0	33.3	**50.0**
date_of_death	66.6	50.0	**57.1**	66.6	50.0	**57.1**	50.0	25.0	33.3	66.6	50.0	**57.1**	66.6	50.0	**57.1**
(per:)parents	37.0	50.0	42.5	42.1	40.0	41.0	37.5	15.0	21.4	34.6	45.0	39.1	40.9	45.0	**42.9**
(org:)alternate_names	20.0	28.5	23.5	18.7	85.7	30.7	20.0	28.5	23.5	16.2	85.7	27.2	19.3	85.7	**31.5**
statesorprovinces_of_residence	50.0	55.5	**52.6**	50.0	44.4	47.0	53.5	44.4	48.5	45.4	55.5	49.9	50.0	44.4	47.0
founded_by	53.8	43.7	48.2	80.0	50.0	61.5	75.0	37.5	50.0	62.5	62.5	62.5	81.8	56.2	**66.6**
children	21.4	27.2	24.0	35.7	45.4	40.0	50.0	9.2	15.5	27.2	27.2	27.2	38.4	45.4	**41.6**
city_of_headquarters	42.8	100.0	59.9	46.1	66.6	54.5	36.3	88.8	51.5	46.6	77.7	58.3	71.4	55.5	**62.5**
siblings	100.0	28.5	**44.4**	100.0	28.5	**44.4**	100.0	14.2	24.9	66.6	28.5	39.9	100.0	28.5	**44.4**
(org:)parents	33.3	33.3	33.3	33.3	66.6	**44.4**	33.3	33.3	33.3	33.3	33.3	33.3	33.3	66.6	**44.4**
Official TAC Scorer (Micro-F_1)	29.3	28.1	28.7	35.5	33.7	34.7	22.7	26.0	24.3	37.5	29.4	33.0	36.9	35.9	**36.4**

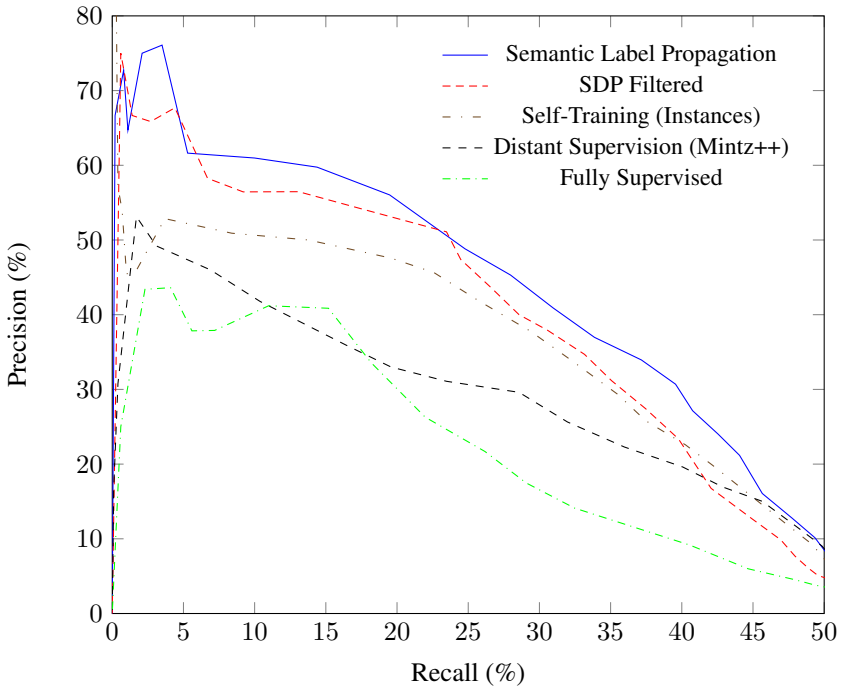

Figure 7: Precision-Recall Graph displaying the output of the TAC KBP evaluation script on different systems, for varying classifier decision thresholds.

One would expect the SDP filtered and fully supervised extractors to attain high precision, but this is not the case for some of the relations. For example, for relation *countries_of_residence* recall of these extractors is higher than recall of the SLP method. However, only those precision and recall scores are shown that correspond to the maximum values for F_1 and while precision could have been higher for these extractors at the cost of lower recall, recall is equally important for this type of evaluation. The SDP filtered and fully supervised extractors are likely to attain high precision values, but this will not compensate for the loss in recall when evaluating F_1 scores. We conclude by noting that the results may also be influenced to peculiarities of the data. Entities chosen by TAC may not always be representative for the majority of persons or organizations in the training data: TAC entities are in many cases more difficult than the average entity from the training set and the most common way of expressing a relationship for these entities might not be present in the test set.

4.6. 2015 TAC KBP Cold Start Slot Filling

The Slot filling task in TAC KBP in 2015 was organized as part of the Cold Start Slot Filling track, where the goal is to search the same document collection to fill in values for specific slots for specific entities, and in a second stage fill slots for answers given during the first stage. In the authors' TAC KBP 2015 submission [65], the ideas presented in this paper were applied, leading to a second place in the Slot Filling Variant. The results showed the influence of a clean training set and the effectiveness of self-training. A top-performing entry was again based on a database system similar to DeepDive [51] and training set filtering using high-precision patterns. We note that the idea of self-training using a first stage high-precision classifier was also included in this system, independently of the work presented in this paper. Some participants successfully used ensembles of neural architectures for relation extraction. However, a selection of our linear classifiers in combination with a careful filtering of distantly supervised training data was shown to outperform these more sophisticated ensembles.

5. Conclusions

In this paper we set out to create high quality training data for relation extractors for automatic knowledge base population systems, while requiring negligible amounts of supervision. To achieve this, we combine the following techniques for the unsupervised generation of training data and manual supervision: (i) *distant supervision (DS):* known relations from an existing knowledge base are used to automatically generate training data, (ii) *feature annotation:* rather than labeling instances, features (e.g., text patterns expressing a relationship) are annotated, selected by means of an active learning criterion based on confidence, and (iii) *semantic feature space representation:* low dimensional vector representations are used to detect additional, semantically related patterns that do not occur in the thus far selected training data, leaving useful patterns undetected otherwise. Thus, we address the problem of noisy training data obtained when using DS alone, by filtering of the training data using high-precision patterns to increase precision (see [53]). After this, to improve recall, we introduce the semi-supervised Semantic Label Propagation method, that allows relaxing the pattern-based filtering of the DS training data by again including weakly supervised items that are sufficiently "similar" to highly confident instances. We found that a simple linear combination of the embeddings of words in a relation pattern is an effective representation when propagating labels from supervised to weakly supervised instances. Tuning a threshold parameter for similarity creates an improved training set for relation extraction.

The main contributions of this paper to the domain of relation extraction and automatic knowledge base population, are (i) the novel methodology of filtering an initial DS training set, where we motivated and demonstrated the effectiveness of an almost negligible manual annotation effort, and (ii) the Semantic Label Propagation model for again expanding the filtered set in order to increase diversity in the training data. We evaluated our classifiers on the knowledge base population task of TAC KBP and showed the competitiveness with respect to established methods that rely on a much heavier annotation cost.

References

[1] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 697–706.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, Dbpedia-a crystallization point for the web of data, Web Semantics: science, services and agents on the world wide web 7 (3) (2009) 154–165.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008, pp. 1247–1250.

[4] T. M. Mitchell, W. W. Cohen, E. R. H. Jr., P. P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. T. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., 2015, pp. 2302–2310.

[5] N. Nakashole, M. Theobald, G. Weikum, Scalable knowledge harvesting with high precision and high recall, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 227–236.

[6] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, S. Soderland, Textrunner: open information extraction on the web, in: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, 2007, pp. 25–26.

[7] J. Fan, D. Ferrucci, D. Gondek, A. Kalyanpur, Prismatic: Inducing knowledge from a large scale lexicalized relation resource, in: Proceedings of the NAACL HLT 2010 first international workshop on formalisms and methodology for learning by reading, Association for Computational Linguistics, 2010, pp. 122–127.

[8] R. Speer, C. Havasi, Representing general relational knowledge in ConceptNet 5, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012, 2012, pp. 3679–3686.

[9] E. Cambria, D. Olsher, D. Rajagopal, SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., 2014, pp. 1515–1521.

[10] S. Poria, E. Cambria, A. F. Gelbukh, F. Bisio, A. Hussain, Sentiment data flow analysis by means of dynamic linguistic patterns, IEEE Comp. Int. Mag. 10 (4) (2015) 26–36.

[11] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 601–610.

[12] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, A. Doan, Building, maintaining, and using knowledge bases: A report from the trenches, in: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, ACM, New York, NY, USA, 2013, pp. 1209–1220.

[13] B. Min, R. Grishman, L. Wan, C. Wang, D. Gondek, Distant supervision for relation extraction with an incomplete knowledge base, in: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, 2013, pp. 777–782.

[14] S. Miller, H. Fox, L. Ramshaw, R. Weischedel, A novel use of statistical parsing to extract information from text, in: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, Association for Computational Linguistics, 2000, pp. 226–233.

[15] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, in: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Association for Computational Linguistics, 2004, p. 22.

[16] E. Boschee, R. Weischedel, A. Zamanian, Automatic information extraction, in: Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA, Citeseer, 2005, pp. 2–4.

[17] J. Jiang, C. Zhai, A systematic exploration of the feature space for relation extraction, in: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA, 2007, pp. 113–120.

[18] A. Sun, R. Grishman, S. Sekine, Semi-supervised relation extraction with large-scale word clustering, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 521–529.

[19] R. C. Bunescu, R. J. Mooney, A shortest path dependency kernel for relation extraction, in: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2005, pp. 724–731.

[20] E. Cambria, B. White, Jumping nlp curves: a review of natural language processing research [review article], Computational Intelligence Magazine, IEEE 9 (2) (2014) 48–57.

[21] E. Agichtein, L. Gravano, Snowball: Extracting relations from large plain-text collections, in: Proceedings of the fifth ACM conference on Digital libraries, ACM, 2000, pp. 85–94.

[22] S. Gupta, C. D. Manning, Spied: Stanford pattern-based information extraction and diagnostics, Proceedings of the ACL 2014 Workshop on Interactive Language Learning, Visualization, and Interfaces (ACL-ILLVI).

[23] M. Surdeanu, H. Ji, Overview of the english slot filling track at the tac2014 knowledge base population evaluation, Proc. Text Analysis Conference (TAC2014).

[24] J. Shin, S. Wu, F. Wang, C. De Sa, C. Zhang, C. Ré, Incremental knowledge base construction using deepdive, Proceedings of the VLDB Endowment 8 (11) (2015) 1310–1321.

[25] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, 2010, pp. 148–163.

[26] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D. S. Weld, 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, Association for Computational Linguistics, 2011, pp. 541–550.

[27] M. Surdeanu, J. Tibshirani, R. Nallapati, C. D. Manning, 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, Association for Computational Linguistics, 2012, pp. 455–465.

[28] G. Angeli, J. Tibshirani, J. Wu, C. D. Manning, Combining distant and partial supervision for relation extraction, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1556–1567.

[29] M. Pershina, B. Min, W. Xu, R. Grishman, Infusion of labeled data into distant supervision for relation extraction, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 732–738.

[30] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, 2009, pp. 1003–1011.

[31] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 71–78.

[32] A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2004, p. 423.

[33] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland, 2014, pp. 2335–2344.

[34] K. Xu, Y. Feng, S. Huang, D. Zhao, Semantic relation classification via convolutional neural networks with simple negative sampling, arXiv preprint arXiv:1506.07650.

[35] H. Adel, B. Roth, H. Schütze, Comparing convolutional neural networks to traditional models for slot filling, arXiv preprint arXiv:1603.05157.

[36] S. Brin, Extracting patterns and relations from the world wide web., Technical Report 1999-65, Stanford InfoLab (November 1999).

[37] C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li, J. Guo, Construction of semantic bootstrapping models for relation extraction, Knowledge-Based Systems 83 (2015) 128–137.

[38] M. Komachi, T. Kudo, M. Shimbo, Y. Matsumoto, Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, pp. 1011–1020.

[39] J. R. Curran, T. Murphy, B. Scholz, Minimising semantic drift with mutual exclusion bootstrapping, Proceedings of the Conference of the Pacific Association for Computational Linguistics (2007) 172–180.

[40] D. S. Batista, B. Martins, M. J. Silva, Semi-supervised bootstrapping of relationship extractors with distributional semantics, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 499–504.

[41] M. Craven, J. Kumlien, Constructing biological knowledge bases by extracting information from text sources, in: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, August 6-10, 1999, Heidelberg, Germany, 1999, pp. 77–86.

[42] I. Augenstein, A. Vlachos, D. Maynard, Extracting relations between non-standard entities using distant supervision and imitation learning (2015) 747–757.

[43] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D. S. Weld, 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, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 541–550.

[44] A. Intxaurrondo, M. Surdeanu, O. L. de Lacalle, E. Agirre, Removing noisy mentions for distant supervision, Procesamiento del lenguaje natural 51 (2013) 41–48.

[45] E. Alfonseca, K. Filippova, J.-Y. Delort, G. Garrido, Pattern learning for relation extraction with a hierarchical topic model, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 54–59.

[46] S. Takamatsu, I. Sato, H. Nakagawa, Reducing wrong labels in distant supervision for relation extraction, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 721–729.

[47] J. Chen, D. Ji, C. L. Tan, Z. Niu, Relation extraction using label propagation based semi-supervised learning, in: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2006, pp. 129–136.

[48] C. Zhang, F. Niu, C. Ré, J. Shavlik, Big data versus the crowd: Looking for relationships in all the right places, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 825–834.

[49] H. Ji, R. Grishman, Knowledge base population: Successful approaches and challenges, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 1148–1158.

[50] G. Angeli, S. Gupta, M. Jose, C. D. Manning, C. Ré, J. Tibshirani, J. Y. Wu, S. Wu, C. Zhang, Stanford's 2014 slot filling systems, TAC KBP.

[51] C. Zhang, Deepdive: A data management system for automatic knowledge base construction, Ph.D. thesis, UW-Madison (2015).

[52] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 287–294.

[53] L. Sterckx, T. Demeester, J. Deleu, C. Develder, Using active learning and semantic clustering for noise reduction in distant supervision, in: 4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014), 2014, pp. 1–6.

[54] G. Druck, B. Settles, A. McCallum, Active learning by labeling features, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, Association for Computational Linguistics, 2009, pp. 81–90.

[55] J. Attenberg, P. Melville, F. Provost, A unified approach to active dual supervision for labeling features and examples, in: In European conference on Machine learning and knowledge discovery in databases, 2010, pp. 40–55.

[56] Z. Harris, Distributional structure, Word 10 (23) (1954) 146–162.

[57] J. H. Martin, D. Jurafsky, Speech and language processing, International Edition.

[58] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR abs/1301.3781.

[59] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1532–1543.

[60] W. Blacoe, M. Lapata, A comparison of vector-based representations for semantic composition, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 546–556.

[61] D. Graff, J. Kong, K. Chen, K. Maeda, English gigaword, Linguistic Data Consortium.

[62] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.

[63] M. Feys, L. Sterckx, L. Mertens, J. Deleu, T. Demeester, C. Develder, Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks, in: 7th Text Analysis Conference, Proceedings, 2014, pp. 1–10.

[64] M. Stevenson, M. A. Greenwood, Comparing information extraction pattern models, in: Proceedings of the Workshop on Information Extraction Beyond The Document, Association for Computational Linguistics, 2006, pp. 12–19.

[65] L. Sterckx, J. Deleu, T. Demeester, C. Develder, Ghent University-IBCN participation in TAC-KBP 2015 cold start task, in: 8th Text Analysis Conference, Proceedings (To Appear), 2015.

[66] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman, Indexing by latent semantic analysis, JASIS 41 (6) (1990) 391–407.