Bootstrapping Distantly Supervised IE using Joint Learning and Small Well-structured Corpora

Lidong Bing[♮,§] **Bhuwan Dhingra**[§] **Kathryn Mazaitis**[§] **Jong Hyuk Park**[§] **William W. Cohen**[§]
[♮]AI Platform Department, Tencent Inc., Shenzhen 518000, China
lyndonbing@tencent.com
[§]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
{bdhingra, krivard, jp1, wcohen}@cs.cmu.edu

Abstract

We propose a framework to improve the performance of distantly-supervised relation extraction, by jointly learning to solve two related tasks: concept-instance extraction and relation extraction. We further extend this framework to make a novel use of document structure: in some small, well-structured corpora, sections can be identified that correspond to relation arguments, and distantly-labeled examples from such sections tend to have good precision. Using these as seeds we extract additional relation examples by applying label propagation on a graph composed of noisy examples extracted from a large unstructured testing corpus. Combined with the soft constraint that concept examples should have the same type as the second argument of the relation, we get significant improvements over several state-of-the-art approaches to distantly-supervised relation extraction, and reasonable extraction performance even with very small set of distant labels.

Introduction

In distantly-supervised information extraction (IE), a knowledge base (KB) of relation or concept instances is used to train an IE system. For example, a set of facts like `sideEffect(meloxicam, stomachBleeding)`, `interactsWith(meloxicam, ibuprofen)`, etc are matched against a corpus, and the matching sentences are then used to generate training data consisting of labeled relation mentions. Distant supervision is less expensive to obtain than directly supervised labels, but produces noisy training data whenever matching errors occur. This causes problems especially when few distant labels are available. Hence distant supervision is often coupled with learning methods that allow for noise, e.g., by introducing latent variables for each entity mention (Hoffmann et al. 2011; Riedel, Yao, and McCallum 2010; Surdeanu et al. 2012); by carefully selecting the entity mentions from contexts likely to include specific KB facts (Wu and Weld 2010); or by careful filtering of the KB strings used as seeds (Movshovitz-Attias and Cohen 2012).

Another recently-introduced approach to reducing the noise in distant supervision combines distant labeling with *label propagation* (LP) (Bing et al. 2015; 2016). Label propagation is a family of graph-based semi-supervised learning (SSL) methods in which "nearby" instances in the graph are encouraged to have similar labels. Depending on the LP method, agreement with seed labels can be imposed as a hard constraint (Zhu, Ghahramani, and Lafferty 2003) or a soft constraint (Lin and Cohen 2010; Talukdar and Cohen 2014). When seed-label agreement is a soft constraint, then LP can be viewed as a way of smoothing the seed labels, so that labels for groups of "similar" instances (i.e., instances nearby in the graph) are upweighted if they agree, and downweighted if they disagree (Bing et al. 2015).

In combining distant supervision with LP, one must build a graph that connects instances that are likely to have the same label. Previous systems have constructed graphs which connect mentions appearing in the same coordinate-list structure—e.g., the underlined noun phrases in "Get medical help if you experience chest pain, weakness, or shortness of breath" (Bing et al. 2015). This approach was shown to improve performance in recognizing instances of certain medical noun-phrase (NP) categories, such as drug names and disease names. An extension of this approach (Bing et al. 2016) learned extract relations, using a more complex graph structure.

This paper presents three new contributions extending this line of work. First, we combine the concept-instance extraction and relation-extraction tasks, in the process greatly simplifying the relation-extraction LP step. The combination of the tasks is simple but effective. In (Bing et al. 2016), relation extraction was performed on an "entity centric" corpus, where each document is primarily concerned with a particular "title entity", and the first argument of each relation is always the title entity: hence relation extraction can be viewed as classification, where an entity mention is labeled with its slot filling role, i.e., its relation to the title entity. The intuition behind combining concept extraction and relation extraction is that relation arguments are often constrained to be of a particular type; for example, the `sideEffect` of a drug is necessarily of the type `symptom`. Hence, incorporating type constraints in relation extraction can improve performance.

The second contribution is a novel use of document structure; in particular, we exploit the fact that in some small, well-structured corpora, sections can be identified that correspond fairly accurately to relation arguments. Figure 1

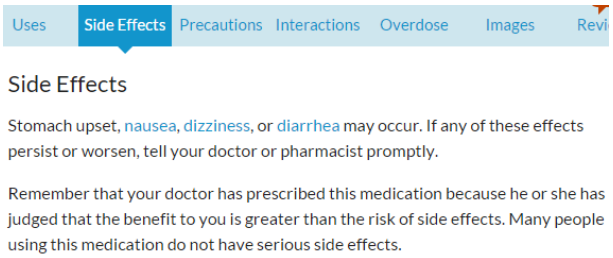


Figure 1: A structured document in WebMD describing the drug meloxicam. All documents in this corpora have the same seven sections.

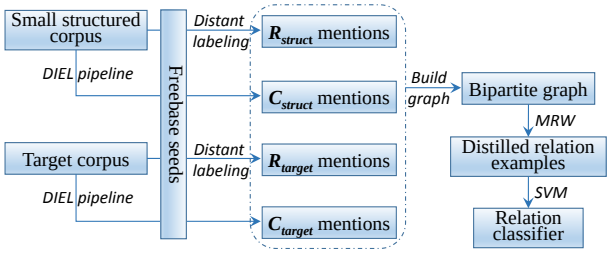


Figure 2: Architecture of DIEJOB.

shows a document from such a structured corpus (discussed later) which contains sections labeled "Side Effects". If "nausea" is distantly labeled as a side effect of `meloxicam` in the "Side Effects" section of this structured document, it is very likely to be a correct mention for the `sideEffect` relation. Used naively, extending a corpus with a small well-structured one needn't lead to improvements, but when combined with LP, we show a consistent and sometimes substantial improvement in performance. We thus illustrate a novel and effective way to make use of a small well-structured corpus, a commonly available resource that is intermediate in structure between a KB and an ordinary text corpus.

The third contribution is experimental. We perform extensive experiments comparing this approach to state-of-the-art distant labeling methods based on latent variables, and show substantial improvements: the relative improvements under F1 measure are from 72% to 110% on one domain, and 22% to 30% on a second domain. Below we present our method in outline and then in detail; present experimental results; discuss related work; and finally conclude.

DIEJOB: *D*istant *IE* by *JO*int *B*ootstrapping

Overview

The architecture for DIEJOB, our system for distantly-supervised relation extraction, is shown in Figure 2. We consider a common case, in which most information is found in relatively unstructured free text, but some smaller corpora exist that are well-structured. DIEJOB thus assumes at least two corpora exist for the domain of interest: a large *target corpus* and a smaller *structured corpus*. Further, it assumes that every document in these two corpora is associated with a particular entity, called *title entity* or *subject entity*. Many widely-used corpora have this structure, including Wikipedia and the authoritative websites we use, DailyMed and WebMD.

From each corpus, DIEJOB produces two types of mention sets: a relation mention set R and a concept mention set C. For the example of Figure 1, R contains a `sideEffect` relation mention for "stomach upset" from the first sentence, and C may contain mentions of the `Symptom` concept, like "stomach upset" and "nausea" from the same sentence. The tail argument values (such as "nausea" in `sideEffect(meloxicam, nausea)`) of a relation are often from a particular unary concept. For example, the sideEffect relation takes instances of Symptom concept as the value range of its second argument.[1] Naively, those concept mentions in C could serve as a source to generate relation examples, but not all concept mentions are relation mentions: e.g., the `Symptom` mentions of "confusion" and "mood changes" from "Symptoms of overdose may include: confusion, mood changes ..." are not mentions of the `sideEffect` relation (or any other relation we currently extract). For the structured corpus, the relation and concept mention sets are referred to as R_s and C_s, and for the target corpus R_t and C_t. Some special treatments (discussed below) are done while preparing R_s and C_s.

After producing R_s, R_t, C_s and C_t, DIEJOB builds a bipartite graph, following prior work (Lin 2012), in which the nodes are either mentions in the four sets, or features of these mentions, with edges between a mention and its features. To distill a cleaner set of relation training examples, DIEJOB performs LP on the bipartite graph. Only the mentions from R_s are used as seed relation examples in this LP stage (because they are more accurate, discussed later).

Finally the distilled relation examples are used to train an ordinary SVM classifier over their extracted features. DIEJOB thus finally learns to classify an unseen mention by the relation which holds between the mention and its corresponding title entity based on features of the mention—a convenient architecture to use for large-scale extraction.

Below we will describe the components of DIEJOB and the experiments in more detail.

Relations and Corpora[2]

Even large curated KBs are often incomplete and the situation is worse in the medical domain where the coverage of large KBs like Freebase is fairly limited. We focus on extracting instances of eight commonly-used relations, defined in Freebase, about drugs and diseases. The drug relations are `usedToTreat`, `conditionsThisMayPrevent`, and `sideEffect`. The concept types of their second arguments are `DiseaseOrMedicalCondition`, `DiseaseOrMedicalCondition`, and `Symptom`, as defined by Freebase. The disease relations are `hasTreatment`, `hasSymptom`, `riskFactor`,

[1] This is also true for general domains: the founder of a company should be a `Person` instance, and its headquarters is usually a `City` instance.

[2] We released some data at: http://curtis.ml.cmu.edu/gnat/ and http://www.wcohen.com

hasCause, and preventionFactor, with corresponding concept types as MedicalTreatment, Symptom, RiskFactor, DiseaseCause, and ConditionPreventionFactor.

We are primarily concerned with extraction from large, authoritative sources. Our target drug corpus, called DailyMed, is downloaded from dailymed.nlm.nih.gov and contains 28,590 XML documents. Our target disease corpus, called WikiDisease, is extracted from a Wikipedia dump of May 2015 and contains 8,596 disease articles. The structured drug corpus[3], called WebMD, contains 2,096 pages collected from www.webmd.com. Each page has the same sections, such as *Uses* and *Side Effects*, corresponding to usedToTreat/conditionsThisMayPrevent and sideEffect relations, respectively. The structured disease corpus, called MayoClinic, contains 1,117 pages collected from www.mayoclinic.org. Each page also has regular sections, such as *Symptoms*, *Causes*, *Risk Factors*, *Treatments/Drugs*, and *Prevention*, corresponding to hasSymptom, hasCause, riskFactor, hasTreatment, and preventionFactor. These corpora are all entity centric, i.e., each page discusses a single entity. [4]

We use GDep (Sagae and Tsujii 2007), a dependency parser trained on GENIA Treebank, to parse the corpora, followed by a simple POS-tag based chunker to extract NPs. We also extract a list (e.g. "stomach upset, nausea, and dizziness") for each coordinating conjunction whose edge label is "NMOD" in the dependency tree. For each NP mention, we extract features (described below) from its context; and for each coordinate list, we extract similar features of the NP chunks. A mention not inside a list is regarded as a singleton list that contains only one item.

Mention Preparation

Relation mention sets, i.e. R_s and R_t, are prepared with distant supervision. The extracted NP mentions are distantly labeled using relation seed triples from Freebase (e.g. sideEffect(meloxican,nausea)). Specifically, we require that the title entity matches the first argument value of the relation, and the NP mention matches the second argument value. To improve the quality of R_s, we also require that the section from which the mention was taken is relevant to the relation; e.g., a mention labeled with the sideEffect relation must appear in a section entitled *Side Effects*. Such a constraint limits the number of labeled mentions in R_s. In the next section, we will show how to extend this small but accurate example set to a larger training set of examples, with reasonable quality.

[3]It is not difficult to find such structured pages in different domains, such as scientist (http://famouschemists.org/, having "Famous For", "Awards", and "Discoveries" sections) and movie (http://www.imdb.com/chart/top, having "Awards", "Plot Summary", etc.)

[4]In fact, entity-centric corpora are common in different domains, such as animal (http://a-z-animals.com/animals/), world heritage (http://whc.unesco.org/en/list), disease gene (https://www.genecards.org/cgi-bin/listdiseasecards.pl, and software (http://stackoverflow.com/tags).

The concept mentions are designed to have high recall with respect to possible argument values for a relation. For each relation r, we generate a set of concept mentions which lie in the range of r's second argument. Following the DIEL system (Bing et al. 2015), we extract concept instances from Freebase as seeds, and extend the seed set using LP in each corpus. Then the coordinate-term lists and singleton lists (NPs) are collected as concept mentions. Thus, we get two concept mention sets: C_s from the structured corpus, and C_t from the target corpus. Note that some mentions in C_s may come from unrelated sections; for instance, C_s for the Symptom concept may contain mentions from the *Overdose* section, which cannot be examples of the sideEffect relation. Therefore, we filter out the mentions in C_s that are not from the appropriate section for this concept (using the mapping from concepts to relations and the mapping from relations to section titles, given in the previous section).

We emphasize that the section-specific processing is *only done on the structured corpus*, i.e. for C_s and R_s. Our target corpora have thousands of section titles, most of which are not related in any way to the relations being extracted. Thus the target relation mentions (R_t) and target concept mentions (C_t) are collected without considering section information.

Relation Label Propagation

With the relation mentions and the concept mentions lying in the range of the corresponding relation, we are able to distill a cleaner set of training relation examples to learn extractors. R_s contains more confident relation examples because of constraints by document structure, but it is limited in size. In contrast, the number of R_t mentions is larger, but they are noisier. In general, the degree to which R_t mentions will be useful may be domain- and corpus-specific. C_s and C_t are generated with respect to the type of the mentions, but not their relationship with the title entity: e.g., a mention in C_t corresponding to the NP "dizziness" would not be associated with the triple sideEffect(meloxican,dizziness); and indeed, dizziness might be a condition treated by, not caused by, the title entity "meloxican". Therefore, C_t itself cannot be directly used as relation examples, however, it can serve as a resource to distill relation examples. In our experiments, R_s mentions are always used as seed relation examples in LP, but we build bipartite propagation graphs with different combinations of the four sets of mentions and study their relative performance.

In total, we have 7 bipartite graphs, each with a different set of mentions from the following combinations: $R_s \cup C_s \cup R_t \cup C_t$, $R_s \cup C_s \cup R_t$, $R_s \cup C_s \cup C_t$, $R_s \cup C_s$, $R_s \cup R_t \cup C_t$, $R_s \cup R_t$, or $R_s \cup C_t$. In a bipartite graph, one set of nodes are mentions, and the other set of nodes are features of mentions. An edge is added between each feature and each mention containing that feature. The edges are TFIDF-weighted (treating the features as words and the mentions as documents).

We use an existing multi-class label propagation method, namely, MultiRankWalk (MRW) (Lin and Cohen 2010), which is a graph-based SSL method related to personalized

PageRank (PPR) (Haveliwala et al. 2003), aka random walk with restart (Tong, Faloutsos, and Pan 2006). MRW can be viewed simply as computing a personalized PageRank vector for each class, each of which is computed using a personalization vector that is initially uniform over the seeds, and finally assigning to each node the class associated with its highest-scoring vector. MRW's final scores depend on the centrality of nodes, as well as their proximity to seeds. The MRW implementation we use is based on ProPPR (Wang, Mazaitis, and Cohen 2013).

Classifier Learning

Given the ranked mentions of these relation labels from the above LP, we pick the top N to train classifiers, which can then be used to classify the entity mentions (singleton lists) and coordinate lists in any document. We use the same feature generator for both mentions and lists. Shallow features include: tokens in the NPs, and character prefixes/suffixes of these tokens; BOW from the sentence containing the NP; and tokens and bigrams from a window around the NPs. From dependency parsing, we find the verb which is closest ancestor of the head of current NP, all modifiers of this verb, and the path to this verb. For lists, the dependency features are computed relative to the head of the list.

We use SVMs (Chang and Lin 2001) and discard singleton features, as well as the most frequent 5% of features (as a stop-wording variant). Specifically, binary classifiers are trained with examples of one relation as the positives, and examples of the other classes as negatives. We also add N general negative examples, randomly picked from those that are not distantly labeled by any relation. A linear kernel and default values for all other parameters are used [5]. A threshold of 0.5 is used to separate positive and negative predictions, and the positive class with the highest probility is finally selected. If a new list or mention is not classified as positive by any classifier, it is predicted as "other".

Experiments

Experimental Comparisons

The first three baselines are distant supervision (DS) systems. They classify each testing NP mention into one of the relation types or "other", using naive matching to the Freebase seed triples as distant supervision. Each sentence in the corpus is processed with the same preprocessing pipeline to detect NPs, which are then labeled with the Freebase seed triples. The features are defined and extracted in the same way as we did for DIEJOB, and binary classifiers are trained with the same method. The first DS baseline, named *DS_Struct*, only uses the section-filtered examples from a structured corpus, i.e. R_s, as training data. The second DS baseline, named *DS_Target*, only uses labeled examples from the target corpus, i.e. R_t. While the third DS baseline, named *DS_Both*, uses examples from both target corpus and structured corpus.

We also compare against two latent variable learners. The first is *MultiR* (Hoffmann et al. 2011) which models each relation mention separately and aggregates their labels using a deterministic OR. The second is *MIML-RE* (Surdeanu et al. 2012) which has a similar structure to *MultiR*, but uses a classifier to aggregate the mention level predictions into an entity pair prediction. We used the publicly available code from the authors[6] for our experiments. Since these methods do not distinguish between structured and unstructured corpora, we used the union of these corpora in our experiments, and the feature set used in the bipartite graph. We found that the performance of these methods varies significantly with the number of negative examples used during training, and hence we tuned these and other parameters[7] directly on the evaluation data, and report their best performance. The last comparison is *Mintz++* (Surdeanu et al. 2012), a distant-supervision baseline which improves on the original model from (Mintz et al. 2009) by training multiple classifiers, and allowing multiple labels per entity pair.

We also compare with DIEBOLDS (Bing et al. 2016), which uses LP on a graph containing entity mention pairs. The graph used by DIEBOLDS is more complex than the mention-feature graph used here, in DIEJOB. One set of vertices correspond to (title-entity, mention-entity) pairs. The other set of vertices are identifiers for coordinate lists: a mention pair is connected with the lists from any document describing the subject, and containing the mention. Additional edges are also introduced based on document structure and BOW context features. DIEBOLDS performs label propagation from the mention pairs distantly labeled with Freebase relation triples.

Experimental Settings and Evaluation Dateset

We extract triples from Freebase as distant labeling seeds in the same way as (Bing et al. 2016) did. Specifically, if the subject of a triple matches with the drug or disease name of a document in a corpus (structured or target) and its object value appears in that document, it is extracted. For the disease domain, we get 2022, 2453, 905, 753, and 164 triples for `hasTreatment`, `hasSymptom`, `riskFactor`, `hasCause`, and `preventionFactor`, respectively. For the drug domain, we get 3112, 315, and 265 triples for `usedToTreat`, `conditionsThisMayPrevent`, and `sideEffect`, respectively.

We have two strategies to pick the top N lists for classifier learning. One strategy picks the top N directly, without distinguishing if they come from the structured corpus or the target corpus. It is referred to as *DIEJOB_Both*. The other strategy picks the top N examples only from the target corpus, and it is referred to as *DIEJOB_Target*. Here our concern is the difference between the feature distributions of the two corpora.

Our evaluation dataset contains 20 manually labeled pages, 10 pages each from the disease corpus WikiDisease and the drug corpus DailyMed. This data was originally generated in (Bing et al. 2016). The annotated text fragments are manually chunked NPs which are the second argument

[5] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[6] http://aiweb.cs.washington.edu/ai/raphaelh/mr/ and http://nlp.stanford.edu/software/mimlre.shtml

[7] Parameters include the number of epochs (for both *MultiR* and *MIML-RE*) and the number of training folds for *MIML-RE*.

Table 1: Extraction results on the evaluation pages. Starred rows are upper bounds on performance

	Disease			Drug		
	P	R	F1	P	R	F1
DS_Struct	0.300	0.300	0.300	0.232	0.072	0.110
DS_Target	0.228	0.335	0.271	0.170	0.188	0.178
DS_Both	0.233	0.353	0.281	0.154	0.175	0.164
DIEBOLDS	0.143	0.372	0.209	0.050	0.435	0.090
MultiR*	0.198	0.333	0.249	0.156	0.138	0.146
Mintz++*	0.192	0.353	0.249	0.177	0.178	0.178
MIML-RE*	0.211	0.360	0.266	0.167	0.160	0.163
DIEJOB_Target	0.231	0.337	0.275	0.299	0.300	0.300
DIEJOB_Both	0.317	0.333	**0.324**	0.327	0.288	**0.306**
DIEJOB_Target*	0.235	0.339	0.277	0.289	0.425	0.344
DIEJOB_Both*	0.317	0.333	0.324	0.282	0.422	0.338

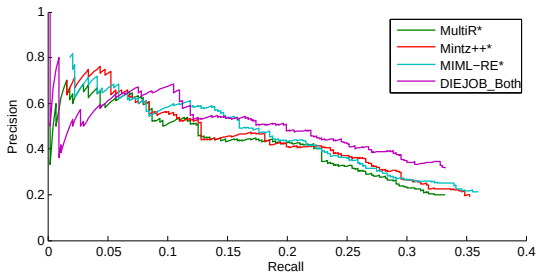
values of any of the eight relations considered here, with the title drug or disease entity of the corresponding document as the relation subject. The evaluation data contains 436 triples for the disease domain and 320 triples for the drug domain. A system's task then is to extract all correct values of the second argument of a given relation from a test document. We evaluate the performance of different systems from an IR perspective: a title entity (i.e., document name) and a relation together act as a query, and the extracted NP strings as retrieval results. For string matching, we employ SecondString with SoftTFIDF as distance metric (Cohen, Ravikumar, and Fienberg 2003) and the match threshold is 0.8 for all compared systems.

Experimental Results

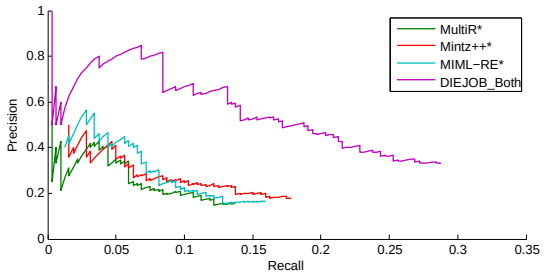
Table 1 shows the microaveraged values of precision, recall and F1 measure. The results for DIEBOLDS are from (Bing et al. 2016). The starred systems are directly tuned on the evaluation data and should be considered as upper bounds on true performance. DIEJOB_Target and DIEJOB_Both are tuned with a tuning dataset (details discussed later). (For the disease domain, DIEJOB_Both and DIEJOB_Both* get the same results, because they use the same parameters, although they are tuned with different data.)

DIEJOB_Both outperforms all the other systems. Compared with MultiR, Mintz++, and MIML-RE, the relative improvements under the F1 measure are 22% to 30% in the disease domain, and 72% to 110% in the drug domain. The precision values of DIEJOB_Both are much higher than previous works. For recall, DIEBOLDS and DIEJOB_Both's performances are comparable to the latent-variable systems on the disease domain and much better on the drug domain. One reason may be DIEJOB's LP step handles the noisy distant examples better than the latent variable models. Another reason is that DIEJOB predicts one label for a coordinate-term list (lists are common in the drug domain), which implicitly coordinates the labels of list items, while MultiR, Mintz++, and MIML-RE break a list into individual items which are predicted separately.

The precision values of DIEBOLDS are much lower than DIEJOB, especially for the drug domain. Unlike DIEJOB, DIEBOLDS builds an LP graph containing all singleton and



(a) Disease domain.



(b) Drug domain.

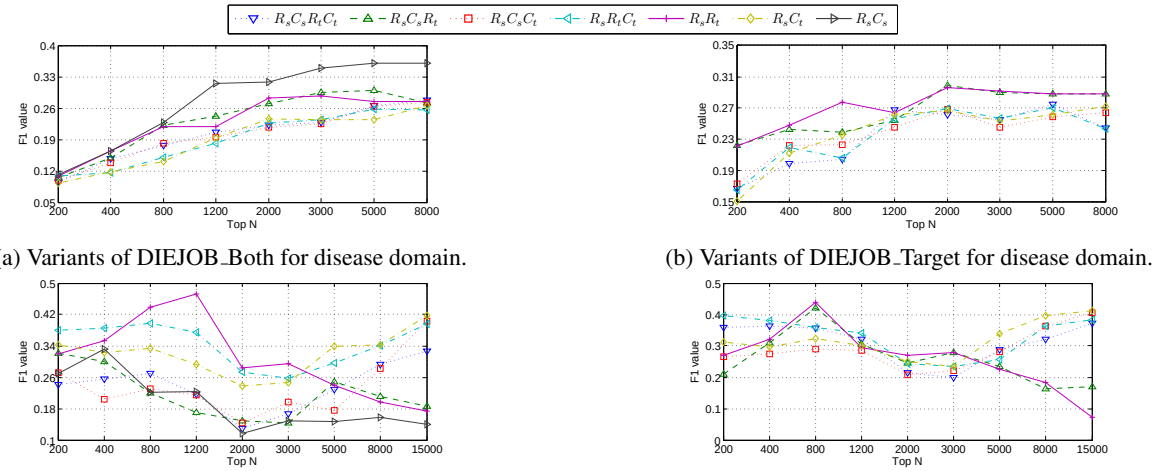
Figure 3: Precision-recall curves.

coordinate lists of noun phrases in the corpus, which introduces many irrelevant examples. DIEBOLDS achieves the highest recall values, but in practice, it is also likely to predict a testing mention as belonging to one of the eight relations, but not "other".

On these tasks, the simple DS baselines' performance is competitive with previous works. One exception is DS_Struct on the drug domain, where the recall is only 0.072. This is perhaps because the total number of examples in R_s for the three drug relations is only 485, which is very small. The precision of DS_Struct is better than DS_Target and DS_Both for both domains, presumably because of the higher quality of the examples in R_s. DS_Both, which naively extends the target corpus with the structured one, does not lead to improvements, but DIEJOB, which uses the structured corpus to modify LP, does improve.

For the disease domain, DIEJOB_Both performs better than DIEJOB_Target, no matter how they are tuned (i.e. on tuning or evaluation data). This shows that the mentions from R_s and C_s of MayoClinic corpus provide good training examples. For the drug domain, DIEJOB_Both and DIEJOB_Target achieve similar results. This may be because DIEJOB_Both is more sensitive to the difference in feature distributions of structured and target corpora, since it uses examples from the structured corpus to learn classifiers as well. Among the four corpora we use, WebMD, MayoClinic, and WikiDisease are written to be readable by a large audience, while DailyMed articles are more difficult in terms of readability: hence the difference between the structured and unstructured corpora is larger in the drug domain.

Precision-recall curves are given in Figure 3. For the drug domain, DIEJOB's precision is consistently better, at the same recall level, than any of the other methods. For the disease domain, our system's precision is generally better after the recall level 0.05.



(a) Variants of DIEJOB_Both for disease domain.

(b) Variants of DIEJOB_Target for disease domain.

(c) Variants of DIEJOB_Both for drug domain.

(d) Variants of DIEJOB_Target for drug domain.

Figure 4: Performance of DIEJOB variants and effect of the parameter N.

Tuning and Ablation Studies

Here we examine the performance of different variants, and the effect of the parameter N. The performance of all graph variants on a tuning dataset (containing 10 labeled pages) is given in Figure 4. Combined with the strategies for picking top N (i.e. DIEJOB_Target and DIEJOB_Both), there are 13 variants: shown in Figures 4a and 4b for disease; Figures 4c and 4d for drug. (Note that DIEJOB_Target does not have the variant R_sC_s, because R_sC_s does not contain any examples from the target corpus.)

For the disease domain, each variant of DIEJOB_Both and DIEJOB_Target performs similarly to its counterpart, on average, DIEJOB_Both is slightly better than DIEJOB_Target. For the drug domain, on average, DIEJOB_Target is better than DIEJOB_Both. One explanation is that the two corpora in the disease domain are more similar, so combining examples from them is more beneficial. However, the effect of such a mixture is negative for drug domain, whose structured and target corpora are more dissimilar.

In Table 1, the reported results of the tuned DIEJOB_Both and DIEJOB_Target for the disease domain are from the variants R_sC_s and $R_sC_sR_t$ respectively, while for the drug domain, both are from R_sR_t. One explanation could be: (1) if the structured corpus is similar to the target corpus, it is better to use DIEJOB_Both, and including examples of the structured corpus (e.g., R_sC_s and $R_sC_sR_t$, both have C_s used) generally performs well with a larger N value; (2) if the structured and target corpora are dissimilar, DIEJOB_Target is better and R_sR_t has an advantage over other variants, as the main focus is distilling good training examples from R_t and a smaller number of top N examples is preferred.

Related Work

To overcome the noise in distantly-labeled examples, (Riedel, Yao, and McCallum 2010) introduced an "at least one" heuristic, where instead of taking all mentions for a pair as correct examples only at least one of them is assumed to express that relation. MultiR (Hoffmann et al. 2011) and MIML-RE (Surdeanu et al. 2012) extend this approach to support multiple relations expressed by different sentences in a bag. Unlike them, DIEJOB improves the quality of training data with a bootstrapping step before feeding the noisy examples into a learner, by using the confident examples from a structured corpus as seeds. The benefit of this step is two-fold: (1) It distills the distantly-labeled examples by propagating labels from good seeds, and downweights the noisy ones; (2) The propagation will walk to more relation examples in the concept mention set that cannot be distantly labeled with triples from knowledge bases.

Document structure was previously explored by (Bing et al. 2016), which used the structure to enrich an LP graph by adding coupling edges between mentions in the same section of particular documents. In this work, we explore the semantic association between section titles and relation arguments. Furthermore, we perform a joint bootstrapping on relation and type mentions to collect training examples with better quality. Technically, the propagation graphs used are different: DIEJOB's graph has carefully produced mention nodes (from those four sets) and their feature nodes, while DIEBOLDS' graph has triple nodes (i.e., subject-NP pairs) and all singleton and coordinate lists of noun phrases of the corpora. Accordingly, their propagation seeds are different: DIEJOB uses confident examples as seeds (labeled from particular sections of a structured corpus) to propagate labels to more examples via feature similarity, while DIEBOLDS directly uses Freebase triples as seeds and propagates labels through edges built from coordinate lists and sections.

In the classic bootstrap learning scheme (Riloff and Jones 1999; Agichtein and Gravano 2000; Bunescu and Mooney 2007), a small number of seed instances are used to extract new patterns from a large corpus, which are then used to extract more instances. Then, in an iterative fashion, new instances are used to extract more patterns. DIEJOB departs from earlier bootstrapping methods in combining label propagation with a standard classification learner, and it can

improve the quality of distant examples and collect new examples simultaneously.

Conclusions

We proposed the DIEJOB framework to generate good examples for distantly-supervised IE. It exploits the document structure of a small well-structured corpus to collect seed relation examples, and it also collects concept mentions that could be the second argument values of relations. DIEJOB then conducts label propagation to find mentions that can be confidently used as training examples to train classifiers for labeling new entity mentions. The experimental results show that this approach consistently and significantly outperforms state-of-the-art approaches, and performs well when few distant labels are available.

Acknowledgments

This work was funded by a grant from Baidu USA and by the NSF under research grant IIS-1250956.

References

Agichtein, E., and Gravano, L. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*.

Bing, L.; Chaudhari, S.; Wang, R. C.; and Cohen, W. W. 2015. Improving distant supervision for information extraction using label propagation through lists. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 524–529.

Bing, L.; Ling, M.; Wang, R.; and Cohen, W. W. 2016. Distant ie by bootstrapping using lists and document structure. *CoRR* abs/1601.00620.

Bunescu, R. C., and Mooney, R. J. 2007. Learning to extract relations from the web using minimal supervision. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Chang, C.-C., and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

Cohen, W. W.; Ravikumar, P.; and Fienberg, S. E. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*.

Haveliwala, T.; Kamvar, S.; Kamvar, A.; and Jeh, G. 2003. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 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.

Lin, F., and Cohen, W. W. 2010. Semi-supervised classification of network data using very few labels. In Memon, N., and Alhajj, R., eds., *ASONAM*, 192–199. IEEE Computer Society.

Lin, F. 2012. *Scalable methods for graph-based unsupervised and semi-supervised learning*. Ph.D. Dissertation, Carnegie Mellon University.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 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: Volume 2-Volume 2*, 1003–1011.

Movshovitz-Attias, D., and Cohen, W. W. 2012. Bootstrapping biomedical ontologies for scientific text using nell. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, 11–19.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 148–163.

Riloff, E., and Jones, R. 1999. Learning Dictionaries for Information Extraction by Multi-level Boot-strapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1044–1049.

Sagae, K., and Tsujii, J. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proceedings of the CoNLL 2007 Shared Task in the Joint Conferences on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07 shared task)*, 1044–1050.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 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*, EMNLP-CoNLL '12, 455–465.

Talukdar, P. P., and Cohen, W. W. 2014. Scaling graph-based semi supervised learning to large number of labels using count-min sketch. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, 940–947.

Tong, H.; Faloutsos, C.; and Pan, J.-Y. 2006. Fast random walk with restart and its applications. In *ICDM*, 613–622. IEEE Computer Society.

Wang, W. Y.; Mazaitis, K.; and Cohen, W. W. 2013. Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2129–2138. ACM.

Wu, F., and Weld, D. S. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 118–127.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of ICML-03, the 20th International Conference on Machine Learning*.