Improving Distant Supervision for Information Extraction Using Label Propagation Through Lists

Lidong Bing[§] **Sneha Chaudhari**[§] **Richard C. Wang**[♮] **William W. Cohen**[§]
[§]Carnegie Mellon University, Pittsburgh, PA 15213
[♮]US Development Center, Baidu USA, Sunnyvale, CA 94089
[§]{lbing@cs, sschaudh@andrew, wcohen@cs}.cmu.edu
[♮]richardwang@baidu.com

Abstract

Because of polysemy, distant labeling for information extraction leads to noisy training data. We describe a procedure for reducing this noise by using label propagation on a graph in which the nodes are entity mentions, and mentions are coupled when they occur in coordinate list structures. We show that this labeling approach leads to good performance even when off-the-shelf classifiers are used on the distantly-labeled data.

1 Introduction

In distantly-supervised information extraction (IE), a knowledge base (KB) of relation or concept instances is used to train an IE system. For instance, a set of facts like *adverseEffectOf(meloxicam, stomachBleeding)*, *interactsWith(meloxicam, ibuprofen)*, might be used to train an IE system that extracts these relations from documents. In distant supervision, instances are first matched against a corpus, and the matching sentences are then used to generate training data consisting of labeled entity mentions. For instance, matching the KB above might lead to labeling passage 1 from Table 1 as support for the fact *adverseEffectOf(, stomachBleeding).*

A weakness of distant supervision is that it produces noisy training data: for instance, matching the adverse effect *weakness* might lead to incorrectly-labeled mention examples. Distant supervision is often coupled with learning methods that allow for this sort of noise by introducing latent variables for each entity mention (e.g., (Hoffmann et al., 2011; Riedel et al., 2010; Surdeanu et al., 2012)); by carefully selecting the entity mentions from contexts likely to include specific KB facts (Wu and Weld, 2010); by careful filtering of the KB strings used as seeds (Movshovitz-Attias and Cohen, 2012); or by making use of named-entity linking methods and co-reference to improve the matching phase of distant learning (Koch et al., 2014).

1. "Avoid drinking alcohol. It may increase your risk of stomach bleeding."
2. "Get emergency medical help if you have chest pain, weakness, shortness of breath, slurred speech, or problems with vision or balance."

Table 1: Passages from a page discussing the drug meloxicam.

Here we explore an alternative approach of ***D**istant **IE** using coordinate-term **L**ists* (DIEL) based on detection of *lists* in text, such as the one illustrated in passage 2 in Table 1. Since list items are usually of the same type, the unambiguous mention *chest pain* here disambiguates the mention *weakness*. Label propagation methods (Zhu et al., 2003; Lin and Cohen, 2010) can be used to exploit this intuition, by propagating the low-confidence labels associated with distance supervision through an appropriate graph.

Here we describe a pipelined system which (1) identifies lists of semantically-related items using lexico-syntactic patterns (2) uses distant supervision, in combination with a label-propagation method, to find entity mentions that can be confidently labeled and (3) from this data, uses ordinary classifier learners to classify entity mentions by their semantic type. We show that this approach outperforms a naive distant-supervision approach.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 524–529,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

2 DIEL: *D*istant *IE* Using Coordinate *L*ists

2.1 Corpus and KB

We consider extending the coverage of Freebase in the medical domain, which is currently fairly limited: e.g., a Freebase snapshot from April 2014 has (after filtering noise with simple rules such as length greater than 60 characters and containing comma) only 4,605 disease instances and 4,383 drug instances, whereas `dailymed.nlm.nih.gov` contains data on over 74k drugs, and `malacards.org` lists nearly 10k diseases. We use a corpus downloaded from `dailymed.nlm.nih.gov` which contains 28,590 XML documents, each of which describes a drug that can be legally prescribed in the United States. We focus here on extracting instances of four semantic types, without explicitly extracting relationships between them.

We used the GDep parser (Sagae and Tsujii, 2007), a dependency parser trained on the GENIA Treebank, to parse this corpus. We used a simple POS-tag based noun-phrase (NP) chunker, and extract a list for each coordinating conjunction that modifies a nominal. For each NP we extract features (described below); and for each identified coordinate-term list, we extract its items, and a similar feature set describing the list.

The extracted lists and their items, as well as entity mentions and their corresponding NPs, can be viewed as a bipartite graph, where one set of vertices are identifiers for the lists and entity mentions, and the other set of vertices are the strings that occur as items of those lists, or as NPs of those mentions. Note that list items are also NPs. A mention can be regarded as a singleton list that contains only one item, and a list can be regarded as a complexus mention that contains a few mentions. If an item is contained by a list, an edge between the item vertex and the list vertex is included in the graph. An example bipartite graph is given in Figure 1, in which there are nine symptoms from three lists and three mentions. Some symptoms are common, such as vomiting, while some others are not, such as epigastric pain.

2.2 Label Propagation

It seems intuitive to assume that if two items cooccur in a coordinate-term list, they are very likely to have the same type, so it seems plausible to use label propagation on this graph to propagate types from NPs with known types (e.g., that match entities in the KB) to lists, and then from lists to NPs, across this graph.

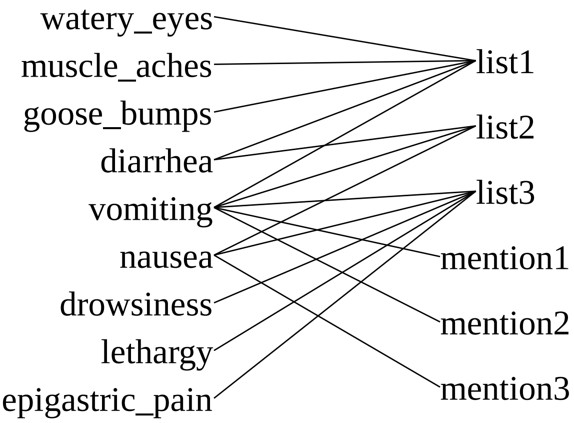

Figure 1: A bipartite graph example.

This can be viewed as semi-supervised learning (SSL) of the NPs that may denote a type (e.g., diseases or adverse effects). We adopt an existing multi-class label propagation method, namely, MultiRankWalk (MRW) (Lin and Cohen, 2010), to handle our task, which is a graph-based SSL related to personalized PageRank (PPR) (Haveliwala et al., 2003) (aka random walk with restart (Tong et al., 2006)). MRW can be described as simply computing one personalized PageRank vector for each class, where each vector is computed using a personalization vector that is uniform over the seeds, and finally assigning to each node the class associated with its highest-scoring vector. MRW's final scores depend on centrality of nodes, as well as proximity to the seeds, and in this respect MRW differs from other label propagation methods (e.g., (Zhu et al., 2003)): in particular, it will not assign identical scores to all seed examples. The MRW implementation we use is based on ProPPR (Wang et al., 2013).

2.3 Classification

One could imagine using the output of MRW to extend a KB directly. However, the process described above cannot be used conveniently to label new documents as they appear. Since this is also frequently a goal, we use the MRW output to train a classifier, which can be then used to classify the entity mentions (singleton lists) and coordinate lists in any new document.

We use the same feature generator for both entity mentions and lists. Shallow features include: tokens in the NPs, and character prefixes/suffixes of these tokens; tokens from the sentence contain-

Run	1	2	3	4	5	6	7	8	9	10	Avg.
DIEL	0.418	0.390	0.404	0.420	0.400	0.397	0.404	0.403	0.409	0.404	0.405
DS-baseline	0.394	0.342	0.400	0.373	0.342	0.297	0.326	0.332	0.330	0.352	0.349
LP-baseline	0.167	0.167	0.162	0.155	0.150	0.159	0.157	0.165	0.167	0.167	0.162
Upper bound	0.617	0.616	0.615	0.619	0.620	0.614	0.618	0.617	0.615	0.618	0.617

Table 2: Recall on the held-out set.

ing the NP; and tokens and bigrams from a window around the NPs. From the dependence parse, we also find the verb which is the closest ancestor of the head of the NP, all modifiers of this verb, and the path to this verb. For a list, the dependency features are computed relative to the head of the list. We used an SVM classifier (Chang and Lin, 2001) and discard singleton features, and also the frequent 5% of all features (as a stop-wording variant). We train a binary classifier on the top N lists (including entity mentions and coordinate lists) of each type, as scored by MRW. A linear kernel and defaults for all other parameters are used. If a new list or mention is not classified as positive by all binary classifiers, it is predicted as “other”.

3 Experimental Results

3.1 Results of Recovering KB

In this experiment, we examine the capability of our approach in recovering KB type instances. The targeted types are diseases, symptoms treated and adverse effects (symptom for short), drugs, and drug ingredients.

3.1.1 Baselines

We implemented a distant-supervision-based baseline (DS-baseline). It attempts to classify each NP in the input corpus into one of the four types or “other” with the training seeds as distance supervision. Each sentence is processed with the same reprocessing pipeline to detect NPs. Then, these NPs are labeled with the training seeds. The features are defined and extracted in the same way as we did for DIEL, and four binary classifiers are trained with the same method. Another baseline is developed with the output of MRW LP (LP-baseline) that contains labeled lists and mentions. Specifically, the labeled coordinate lists are broken into items each of which has the list class, and evaluation is conducted with these items together with the labeled mentions as positive predictions.

3.1.2 Settings

We extracted the seeds of these types from Freebase, and got 4,605, 1,244, 4,383, and 4,066 instances, respectively. The seeds are split into development set and held-out evaluation set. The development set is further split into a training set and a validating set in the ratio of 4:1. The validating set will be used in the next subsection to validate different parameter settings, and the training set is used in this experiment as MRW seeds and the distant supervision of DS-baseline.

For getting the development set, the polysemous instances (i.e., “headache”, belonging to multiple classes: disease and symptom) are discarded since such instances will bring in ambiguity to the training examples of DS-baseline and MRW LP. After that, we randomly take half of the single-type instances as development set, and the remaining single-type instances together with the polysemous instances are used as held-out set. We report the performance of 10 runs, and each run has its own randomly generated training set (containing 1,980 diseases, 310 symptoms, 1,066 drugs, and 911 ingredients on average) and held-out set (containing 2,130 diseases, 857 symptoms, 3,051 drugs, and 2,927 ingredients on average). For DS-baseline, 35,689 training examples are collected on average for each run. For evaluation metric, we calculate recall on the instances in the held-out set.[1]

3.1.3 Results

DIEL can classify both NPs and coordinate lists, and the lists are also broken into items for recall calculation. The training examples of DIEL are prepared with the top 20,000 lists of each type, as scored by MRW with the training seeds. After removing the multiple-class ones, we obtained, on average, 58,614 training examples for each run.

The results are given in Table 2. DIEL outperforms the baselines in all runs. It shows that

[1] Because the outputs contain many correct instances that are not in Freebase, it is not suitable to calculate precision.

		2.5%		7.5%		12.5%		25%		50%		75%		100%	
	N	P	R	P	R	P	R	P	R	P	R	P	R	P	R
DIEL	1,000	0.581	0.119	0.680	0.150	0.706	0.154	0.754	0.216	0.768	0.277	0.749	0.308	0.769	0.331
	2,000	0.587	0.168	0.675	0.183	0.705	0.192	0.746	0.245	0.760	0.292	0.747	0.317	0.777	0.347
	4,000	0.551	0.205	0.661	0.225	0.682	0.254	0.726	0.295	0.763	0.334	0.763	0.356	0.766	0.375
	6,000	0.500	0.205	0.636	0.244	0.650	0.281	0.680	0.314	0.757	0.362	0.758	0.380	0.768	0.392
	8,000	0.518	0.219	0.615	0.251	0.641	0.300	0.662	0.331	0.727	0.393	0.764	0.407	0.762	0.409
	10,000	0.520	0.219	0.610	0.260	0.643	0.309	0.664	0.349	0.708	0.406	0.753	0.436	0.765	0.429
	20,000	0.520	0.216	0.590	0.277	0.634	0.335	0.661	0.375	0.688	0.435	0.723	0.473	0.746	0.494
	30,000	0.520	0.216	0.584	0.284	0.624	0.341	0.649	0.392	0.688	0.454	0.717	0.486	0.737	0.496
LP-baseline		0.442	0.024	0.604	0.074	0.673	0.108	0.722	0.134	0.797	0.173	0.838	0.187	0.855	0.212
DS-baseline	Training NP#	678		2,442		5,983		9,831		18,230		27,549		35,689	
	Performance	0.317	0.139	0.384	0.161	0.643	0.244	0.714	0.252	0.747	0.322	0.767	0.347	0.742	0.351

Table 4: The classification performance of the three methods with different parameters.

Type	Disease	Drug	Ingredient	Symptom
DIEL	0.369	**0.312**	**0.489**	**0.555**
DS-baseline	**0.375**	0.300	0.408	0.224
LP-baseline	0.185	0.101	0.117	0.453
Upper bound	0.445	0.652	0.700	0.697

Table 3: Recall on individual types.

our result is consistently better. The reason is twofold. First, DIEL can avoid the effect of noisy training data by disambiguation with the coordinate relation in the list, so that the training examples are of high quality. Second, with label propagation, we have a larger number of training examples, which helps the recall. Compared with DS-baseline, DIEL's performance is more stable in different runs. It is because DS-baseline suffers from the noisy training data and training seed sets of different runs may bring in different levels of noisy data. Thus, its run 3 achieves 0.400, while run 6 only achieves 0.297. We also examined the upper bound recall that a system can achieve on our corpus. The results are given in the last row of Table 2. On average, the best performance of a system can achieve is 0.617.

The results for individual types are given in Table 3. DIEL and DS-baseline achieve similar results for disease and drug. Especially, both systems cover more than 80% of the held-out disease instances that exist in the corpus. DS-baseline performs poorly for symptom. The reason is that symptom instances are more ambiguous than other types, and they lead to more incorrectly-labeled mention examples. LP-baseline achieves an encouraging recall for symptom, which shows that coordinate lists are very helpful for disambiguating those symptom mentions.

3.2 Classification Results and Parameters

We present another experiment to examine the precision of the systems, and investigate the effect of training size and top N numbers on the results.

3.2.1 Setting

The evaluation data is generated with the validating set of each run. Specifically, for DIEL and LP-baseline, the evaluation data is prepared with the top 500 lists (singleton and coordinate lists) of each type, as scored by MRW with the validating instances as seeds. 2,000 testing examples are collected in total since no multiple-class ones are found. Manual checking on 200 random examples shows that 99% of them are correct. The systems are evaluated by their performance for classifying these example lists. DIEL predicts the class of a list with its feature vector, while LP-baseline determines the class of a list by checking its predicted class in the result of MRW with the corresponding training instances as seeds. For DS-baseline, the testing NP examples are obtained by distant labeling with validating instances, and on average 8,270 examples are collected for each run.

3.2.2 Results

The precision and recall are given in Table 4. For each system, different portions of the training set are used to train the system, as shown in the first row. For DIEL, the top N number varies for generating training examples, as shown in the second column. F1 values of DIEL with different settings are given in Figure 2, and F1 comparison of three systems is given in Figure 3. All results are the average of 10 runs. Each run has its own randomly generated development set, which is split into training set and validating set.

In general, for all systems, larger number of training seeds leads to better performance. For

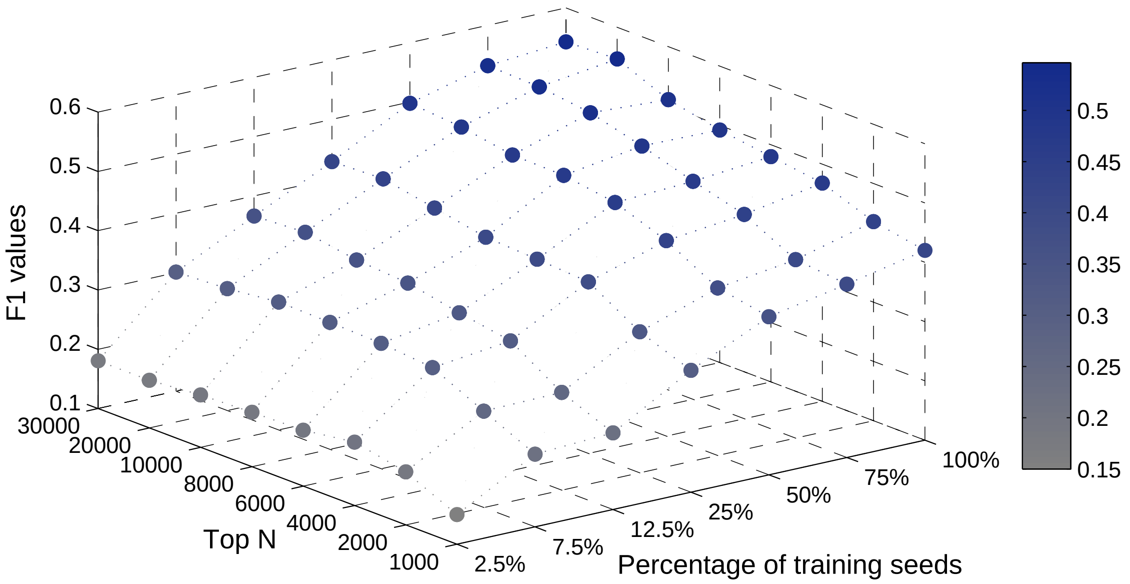

Figure 2: F1 values of DIEL under different settings.

DIEL, smaller N values achieve higher precision, but lower recall. For smaller seed numbers, the precision value is more sensitive to N. This is because the quality of training examples drops faster compared with that from larger seed numbers. For larger seeds numbers, the recall values are improved more significantly when the N value is larger. The reported results of DIEL in the previous experiment are obtained with top 20,000 examples from 100% seeds as training data, since this setting achieves the highest F1 value as shown in Figure 2.

For the DS-baseline, the number of training NPs obtained with different portions of the training set is given in the penultimate row. The recall values of this baseline are low. The reason is that it only uses the training examples that are distantly labeled with training seeds, thus, the trained classifier may not have good generalization on the testing examples labeled with validating seeds. In addition, its performance is more sensitive on the amount of training data. When the percentage is lower than 25%, its precision and recall drop significantly. Its F1 values are 0.381, 0.399 and 0.402 for 50%, 75%, and 100%, respectively. LP-baseline achieves the highest precision when using all training instances. It shows that MRW does label the testing lists very accurately in condition that the lists are traversed in the propagation with the training instances as seeds. However, its recall is much lower than DIEL. It is because, with the training seeds, MRW cannot effectively walk to testing lists that are generated with the validating set, having no intersection with the training set.

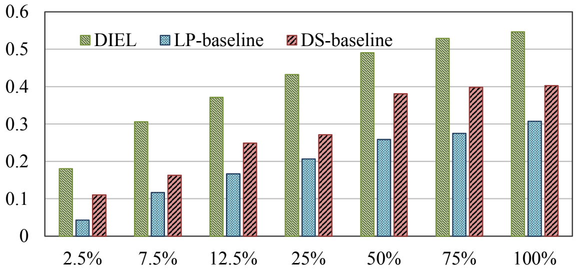

Figure 3: F1 comparison under different portions of the training seeds. For DIEL, top N = 20,000.

4 Conclusions

We explored an alternative approach to distant supervision, based on detection of lists in text, to overcome the weakness of distant supervision resulted by noisy training data. It uses distant supervision and label propagation to find mentions that can be confidently labeled, and uses them to train classifiers to label more entity mentions. The experimental results show that this approach consistently and significantly outperforms a naive distant-supervision approach.

Acknowledgments

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

References

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

Taher Haveliwala, Sepandar Kamvar, Ar Kamvar, and Glen Jeh. 2003. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University.

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

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. 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*, pages 1891–1901.

Frank Lin and William W. Cohen. 2010. Semi-supervised classification of network data using very few labels. In Nasrullah Memon and Reda Alhajj, editors, *ASONAM*, pages 192–199. IEEE Computer Society.

Dana Movshovitz-Attias and William W. Cohen. 2012. Bootstrapping biomedical ontologies for scientific text using nell. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, pages 11–19, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

K. Sagae and J. Tsujii. 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)*, pages 1044–1050.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 455–465, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*, pages 613–622. IEEE Computer Society.

William Yang Wang, Kathryn Mazaitis, and William W Cohen. 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*, pages 2129–2138. ACM.

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

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