Unsupervised Relation Discovery with Sense Disambiguation

Limin Yao **Sebastian Riedel** **Andrew McCallum**
Department of Computer Science
University of Massachusetts, Amherst
{lmyao,riedel,mccallum}@cs.umass.edu

Abstract

To discover relation types from text, most methods cluster shallow or syntactic patterns of relation mentions, but consider only one possible sense per pattern. In practice this assumption is often violated. In this paper we overcome this issue by inducing clusters of pattern senses from feature representations of patterns. In particular, we employ a topic model to partition entity pairs associated with patterns into sense clusters using local and global features. We merge these sense clusters into semantic relations using hierarchical agglomerative clustering. We compare against several baselines: a generative latent-variable model, a clustering method that does not disambiguate between path senses, and our own approach but with only local features. Experimental results show our proposed approach discovers dramatically more accurate clusters than models without sense disambiguation, and that incorporating global features, such as the document theme, is crucial.

1 Introduction

Relation extraction (RE) is the task of determining semantic relations between entities mentioned in text. RE is an essential part of information extraction and is useful for question answering (Ravichandran and Hovy, 2002), textual entailment (Szpektor et al., 2004) and many other applications.

A common approach to RE is to assume that relations to be extracted are part of a predefined ontology. For example, the relations are given in knowledge bases such as Freebase (Bollacker et al., 2008) or DBpedia (Bizer et al., 2009). However, in many applications, ontologies do not yet exist or have low coverage. Even when they do exist, their maintenance and extension are considered to be a substantial bottleneck. This has led to considerable interest in unsupervised relation discovery (Hasegawa et al., 2004; Banko and Etzioni, 2008; Lin and Pantel, 2001; Bollegala et al., 2010; Yao et al., 2011). Here, the relation extractor simultaneously discovers facts expressed in natural language, and the ontology into which they are assigned.

Many relation discovery methods rely exclusively on the notion of either shallow or syntactic patterns that appear between two named entities (Bollegala et al., 2010; Lin and Pantel, 2001). Such patterns could be sequences of lemmas and Part-of-Speech tags, or lexicalized dependency paths. Generally speaking, relation discovery attempts to cluster such patterns into sets of equivalent or similar meaning. Whether we use sequences or dependency paths, we will encounter the problem of polysemy. For example, a pattern such as “A beat B” can mean that person A wins over B in competing for a political position, as pair “(Hillary Rodham Clinton, Jonathan Tasini)” in “Sen Hillary Rodham Clinton beats rival Jonathan Tasini for Senate.” It can also indicate that an athlete A beat B in a sports match, as pair “(Dmitry Tursunov, Andy Roddick)” in “Dmitry Tursunov beat the best American player Andy Roddick.” Moreover, it can mean “physically beat” as pair “(Mr. Harris, Mr. Simon)” in “On Sept. 7, 1999, Mr. Harris fatally beat Mr. Simon.” This is known as *polysemy*. If we work with patterns alone, our extractor will not be able to differentiate between these cases.

Most previous approaches do not explicitly address this problem. Lin and Pantel (2001) assumes only one sense per path. In (Pantel et al., 2007), they augment each relation with its selectional preferences, i.e. fine-grained entity types of two arguments, to handle polysemy. However, such fine grained entity types come at a high cost. It is difficult to discover a high-quality set of fine-grained entity types due to unknown criteria for developing such a set. In particular, the optimal granularity of entity types depends on the particular pattern we consider. For example, a pattern like “A beat B” could refer to A winning a sports competition against B, or a political election. To differentiate between these senses we need types such as “Politician” or “Athlete”. However, for “A, the parent of B” we only need to distinguish between persons and organizations (for the case of the sub-organization relation). In addition, there are senses that just cannot be determined by entity types alone: Take the meaning of “A beat B” where A and B are both persons; this could mean A physically beats B, or it could mean that A defeated B in a competition.

In this paper we address the problem of polysemy, while we circumvent the problem of finding fine-grained entity types. Instead of mapping entities to fine-grained types, we directly induce pattern senses by clustering feature representations of pattern contexts, i.e. the entity pairs associated with a pattern. This allows us to employ not only local features such as words, but also global features such as the document and sentence themes.

To cluster the entity pairs of a single relation pattern into senses, we develop a simple extension to Latent Dirichlet Allocation (Blei et al., 2003). Once we have our pattern senses, we merge them into clusters of different patterns with a similar sense. We employ hierarchical agglomerative clustering with a similarity metric that considers features such as the entity arguments, and the document and sentence themes.

We perform experiments on New York Times articles and consider lexicalized dependency paths as patterns in our data. In the following we shall use the term path and pattern exchangeably. We compare our approach with several baseline systems, including a generative model approach, a clustering method that does not disambiguate between senses, and our approach with different features. We perform both automatic and manual evaluations. For automatic evaluation, we use relation instances in Freebase as ground truth, and employ two clustering metrics, pairwise F-score and B^3 (as used in coference). Experimental results show that our approach improves over the baselines, and that using global features achieves better performance than using entity type based features. For manual evaluation, we employ a set intrusion method (Chang et al., 2009). The results also show that our approach discovers relation clusters that human evaluators find coherent.

2 Our Approach

We induce pattern senses by clustering the entity pairs associated with a pattern, and discover semantic relations by clustering these sense clusters. We represent each pattern as a list of entity pairs and employ a topic model to partition them into different sense clusters using local and global features. We take each sense cluster of a pattern as an atomic cluster, and use hierarchical agglomerative clustering to organize them into semantic relations. Therefore, a semantic relation comprises a set of sense clusters of patterns. Note that one pattern can fall into different semantic relations when it has multiple senses.

2.1 Sense Disambiguation

In this section, we discuss the details of how we discover senses of a pattern. For each pattern, we form a clustering task by collecting all entity pairs the pattern connects. Our goal is to partition these entity pairs into sense clusters. We represent each pair by the following features.

Entity names: We use the surface string of the entity pair as features. For example, for pattern “A play B”, pairs which contain B argument “Mozart” could be in one sense, whereas pairs which have “Mets” could be in another sense.

Words: The words between and around the two entity arguments can disambiguate the sense of a path. For example, “A’s parent company B” is different from “A’s largest company B” although they share the same path “A’s company B”. The former describes the sub-organization relationship between two companies, while the latter describes B as the largest company in a location A. The two words to the left of the source argument, and to the right of the destination argument also help sense discovery. For example, in “Mazurkas played by Anna Kijanowska, pianist”, “pianist” tells us pattern “A played by B” takes the “music” sense.

Document theme: Sometimes, the same pattern can express different relations in different documents, depending on the document’s theme. For instance, in a document about politics, “A defeated B” is perhaps about a politician that won an election against another politician. While in a document about sports, it could be a team that won against another team in a game, or an athlete that defeated another athlete. In our experiments, we use the meta-descriptors of a document as side information and train a standard LDA model to find the theme of a document. See Section 3.1 for details.

Sentence theme: A document may cover several themes. Moreover, sometimes the theme of a document is too general to disambiguate senses. We therefore also extract the theme of a sentence as a feature. Details are in 3.1.

We call entity name and word features local, and the two theme features global.

We employ a topic model to discover senses for each path. Each path p_i forms a document, and it contains a list of entity pairs co-occurring with the path in the tuples. Each entity pair is represented by a list of features f_k as we described. For each path, we draw a multinomial distribution θ over topics/senses. For each feature of an entity pair, we draw a topic/sense from θ_{p_i}. Formally, the generative process is as follows:

$$\begin{aligned} \theta_{p_i} &\sim Dirichlet(\alpha) \\ \phi_z &\sim Dirichlet(\beta) \\ z_e &\sim \text{Multinomial}(\theta_{p_i}) \\ f_k &\sim \text{Multinomial}(\phi_{z_e}) \end{aligned}$$

Assume we have m paths and l entity pairs for each path. We denote each entity pair of a path as $e(p_i) = (f_1, \ldots, f_n)$. Hence we have:

$$\begin{aligned} &P(e_1(p_i), e_2(p_i), \ldots, e_l(p_i)|z_1, z_2, \ldots, z_l) \\ &\qquad = \prod_{j=1}^{l}\prod_{k=1}^{n} p(f_k|z_j)p(z_j) \end{aligned}$$

We assume the features are conditionally independent given the topic assignments. Each feature is generated from a multinomial distribution ϕ. We use Dirichlet priors on θ and ϕ. Figure 1 shows the graphical representation of this model.

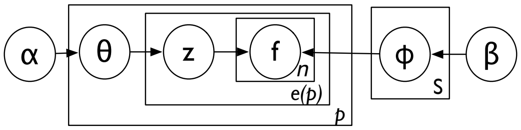

Figure 1: Sense-LDA model.

This model is a minor variation on standard LDA and the difference is that instead of drawing an observation from a hidden topic variable, we draw multiple observations from a hidden topic variable. Gibbs sampling is used for inference. After inference, each entity pair of a path is assigned to one topic. One topic is one sense. Entity pairs which share the same topic assignments form one sense cluster.

2.2 Hierarchical Agglomerative Clustering

After discovering sense clusters of paths, we employ hierarchical agglomerative clustering (HAC) to discover semantic relations from these sense clusters. We apply the complete linkage strategy and take cosine similarity as the distance function. The cutting threshold is set to 0.1.

We represent each sense cluster as one vector by summing up features from each entity pair in the cluster. The weight of a feature indicates how many entity pairs in the cluster have the feature. Some features may get larger weights and dominate the cosine similarity. We down-weigh these features. For example, we use binary features for word “defeat” in sense clusters of pattern “A defeat B”. The two theme features are extracted from generative models, and each is a topic number.

Our approach produces sense clusters for each path and semantic relation clusters of the whole data. Table 1 and 2 show some example output.

3 Experiments

We carry out experiments on New York Times articles from years 2000 to 2007 (Sandhaus, 2008). Following (Yao et al., 2011), we filter out noisy documents and use natural language packages to annotate the documents, including NER tagging (Finkel et al., 2005) and dependency parsing (Nivre et al., 2004). We extract dependency paths for each pair of named entities in one sentence. We use their lemmas

Path	20:sports	30:entertainment	25:music/art
A play B	Americans, Ireland Yankees, Angels Ecuador, England Redskins, Detroit Red Bulls, F.C. Barcelona	Jean-Pierre Bacri, Jacques Rita Benton, Gay Head Dance Jeanie, Scrabble Meryl Streep, Leilah Kevin Kline, Douglas Fairbanks	Daniel Barenboim, recital of Mozart Mr. Rose, Ballade Gil Shaham, Violin Romance Ms. Golabek, Steinways Bruce Springsteen, Saints
doc theme	sports	music books television	music theater
sen theme	game yankees	theater production book film show	music reviews opera
lexical words	beat victory num-num won	played plays directed artistic	director conducted production
entity names	-	r:theater	r:theater r:hall r:york l:opera

Table 1: Example sense clusters produced by sense disambiguation. For each sense, we randomly sample 5 entity pairs. We also show top features for each sense. Each row shows one feature type, where “num” stands for digital numbers, and prefix “l:” for source argument, prefix “r:” for destination argument. Some features overlap with each other. We manually label each sense for easy understanding. We can see the last two senses are close to each other. For two theme features, we replace the theme number with the top words. For example, the document theme of the first sense is Topic30, and Topic30 has top words “sports”.

relation	paths
entertainment	A, who play B:30; A play B:30; star A as B:30

sports	lead A to victory over B:20; A play to B:20; A play B:20; A’s loss to B:20; A beat B:20; A trail B:20;
	A face B:26; A hold B:26; A play B:26; A acquire (X) from B:26; A send (X) to B:26;

politics	A nominate B:39; A name B:39; A select B:39; A name B:42; A select B:42;
	A ask B:42; A choose B:42; A nominate B:42; A turn to B:42;

law	A charge B:39; A file against B:39; A accuse B:39; A sue B:39

Table 2: Example semantic relation clusters produced by our approach. For each cluster, we list the top paths in it, and each is followed by “:number”, indicating its sense obtained from sense disambiguation. They are ranked by the number of entity pairs they take. The column on the left shows sense of each relation. They are added manually by looking at the sense numbers associated with each path.

for words on the dependency paths. Each entity pair and the dependency path which connects them form a tuple.

We filter out paths which occur fewer than 200 times and use some heuristic rules to filter out paths which are unlikely to represent a relation, for example, paths in with both arguments take the syntactic role “dobj” (direct objective) in the dependency path. In such cases both arguments are often part of a coordination structure, and it is unlikely that they are related. In summary, we collect about one million tuples, 1300 patterns and half million named entities. In terms of named entities, the data is very sparse. On average one named entity occurs four times.

3.1 Feature Extraction

For the entity name features, we split each entity string of a tuple into tokens. Each token is a feature. The source argument tokens are augmented with prefix “l:”, and the destination argument tokens with prefix “r:”. We use tokens to encourage overlap between different entities.

For the word features, we extract all the words between the two arguments, removing stopwords and the words with capital letters. Words with capital letters are usually named entities, and they do not tend to indicate relations. We also extract neighboring words of source and destination arguments. The two words to the left of the source argument are added with prefix “lc:”. Similarly the two words to the right of the destination arguments are added with prefix “rc:”.

Each document in the NYT corpus is associated with many descriptors, indicating the topic of the document. For example, some documents are labeled as “Sports”, “Dallas Cowboys”, “New York Giants”, “Pro Football” and so on. Some are labeled as “Politics and Government”, and “Elections”. We shall extract a theme feature for each document from these descriptors. To this end we interpret the descriptors as words in documents, and train a standard LDA model based on these documents. We pick the most frequent topic as the theme of a document.

We also train a standard LDA model to obtain the theme of a sentence. We use a bag-of-words representation for a document and ignore sentences from which we do not extract any tuples. The LDA model assigns each word to a topic. We count the occurrences of all topics in one sentence and pick the most frequent one as its theme. This feature captures the intuition that different words can indicate the same sense, for example, “film’”, “show”, “series” and “television” are about “entertainment”, while “coach”, “game”, “jets”, “giants” and “season” are about “sports”.

3.2 Sense clusters and relation clusters

For the sense disambiguation model, we set the number of topics (senses) to 50. We experimented with other numbers, but this setting yielded the best results based on our automatic evaluation measures. Note that a path has a multinomial distribution over 50 senses but only a few senses have non-zero probabilities.

We look at some sense clusters of paths. For path “A play B”, we examine the top three senses, as shown in Table 1. The last two senses “entertainment” and “music” are close. Randomly sampling some entity pairs from each of them, we find that the two sense clusters are precise. Only 1% of pairs from the sense cluster “entertainment” should be assigned to the “music” sense. For the path “play A in B” we discover two senses which take the most probabilities: “sports” and “art”. Both clusters are precise. However, the “sports” sense may still be split into more fine-grained sense clusters. In “sports”, 67% pairs mean “play another team in a location” while 33% mean “play another team in a game”.

We also closely investigate some relation clusters, shown in Table 2. Both the first and second relation contain path “A play B” but with different senses. For the second relation, most paths state “play” relations between two teams, while a few of them express relations of teams acquiring players from other teams. For example, the entity pair ”(Atlanta Hawks, Dallas Mavericks)” mentioned in sentence ”The Atlanta Hawks acquired point guard Anthony Johnson from the Dallas Mavericks.” This is due to that they share many entity pairs of team-team.

3.3 Baselines

We compare our approach against several baseline systems, including a generative model approach and variations of our own approach.

Rel-LDA: Generative models have been successfully applied to unsupervised relation extraction (Rink and Harabagiu, 2011; Yao et al., 2011). We compare against one such model: An extension to standard LDA that falls into the framework presented by Yao et al. (2011). Each document consists of a list of tuples. Each tuple is represented by features of the entity pair, as listed in 2.1, and the path. For each document, we draw a multinomial distribution over relations. For each tuple, we draw a relation topic and independently generate all the features. The intuition is that each document discusses one domain, and has a particular distribution over relations.

In our experiments, we test different numbers of relation topics. As the number goes up, precision increases whereas recall drops. We report results with 300 and 1000 relation topics.

One sense per path (HAC): This system uses only hierarchical clustering to discover relations, skipping sense disambiguation. This is similar to DIRT (Lin and Pantel, 2001). In DIRT, each path is represented by its entity arguments. DIRT calculates distributional similarities between different paths to find paths which bear the same semantic relation. It does not employ global topic model features extracted from documents and sentences.

Local: This system uses our approach (both sense clustering with topic models and hierarchical clustering), but without global features.

Local+Type This system adds entity type features to the previous system. This allows us to compare performance of using global features against entity type features. To determine entity types, we link named entities to Wikipedia pages using the Wikifier (Ratinov et al., 2011) package and extract categories from the Wikipedia page. Generally Wikipedia provides many types for one entity. For example, “Mozart” is a *person*, *musician*, *pianist*, *composer*, and *catholic*. As we argued in Section 1, it is difficult to determine the right granularity of the entity types to use. In our experiments, we use all of them as features. In hierarchical clustering, for each sense cluster of a path, we pick the most frequent entity type as a feature. This approach can be seen as a proxy to ISP (Pantel et al., 2007), since selectional preferences are one way of distinguishing multiple senses of a path.

Our Approach+Type This system adds Wikipedia entity type features to our approach. The Wikipedia feature is the same as used in the previous system.

4 Evaluations

4.1 Automatic Evaluation against Freebase

We evaluate relation clusters discovered by all approaches against Freebase. Freebase comprises a large collection of entities and relations which come from varieties of data sources, including Wikipedia infoboxes. Many users also contribute to Freebase by annotating relation instances. We use coreference evaluation metrics: pairwise F-score and B^3 (Bagga and Baldwin, 1998). Pairwise metrics measure how often two tuples which are clustered in one semantic relation are labeled with the same Freebase label. We evaluate approximately 10,000 tuples which occur in both our data and Freebase. Since our system predicts fine-grained clusters comparing against Freebase relations, the measure of recall is underestimated. The precision measure is more reliable and we employ F-0.5 measure, which places more emphasis on precision.

Matthews correlation coefficient (MCC) (Baldi et al., 2000) is another measure used in machine learning, which takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used when the classes are of very different sizes. In our case, the true negative number is 100 times larger than the true positive number. Therefor we also employ MCC, calculated as

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

The MCC score is between -1 and 1. The larger the better. In perfect predictions, FP and FN are 0, and the MCC score is 1. A random prediction results in score 0.

Table 3 shows the results of all systems. Our approach achieves the best performance in most measures. Without using sense disambiguation, the performance of hierarchical clustering decreases significantly, losing 17% in precision in the pairwise measure, and 15% in terms of B^3. The generative model approach with 300 topics achieves similar precision to the hierarchical clustering approach. With more topics, the precision increases, however, the recall of the generative model is much lower than those of other approaches. We also show the results of our approach without global document and sentence theme features (Local). In this case, both precision and recall decrease. We compare global features (Our approach) against Wikipedia entity type features (Local+Type). We see that using global features achieves better performance than using entity type based features. When we add entity type features to our approach, the performance does not increase. The entity type features do not help much is due to that we cannot determine which particular type to choose for an entity pair. Take pair "(Hillary Rodham Clinton, Jonathan Tasini)" as an example, choosing *politician* for both arguments instead of *person* will help.

We should note that these measures provide comparison between different systems although they are not accurate. One reason is the following: some relation instances should have multiple labels but they have only one label in Freebase. For example, instances of a relation that a person "was born in" a country could be labeled as "/people/person/place_of_birth" and as "/people/person/nationality". This decreases the pairwise precision. Further discussion is in Section 4.3.

4.2 Path Intrusion

We also evaluate coherence of relation clusters produced by different approaches by creating path intrusion tasks (Chang et al., 2009). In each task, some paths from one cluster and an intruding path from another are shown, and the annotator's job is to identify one single path which is out of place. For each path, we also show the annotators one example sentence. Three graduate students in natural language processing annotate intruding paths. For disagreements, we use majority voting. Table 4 shows one example intrusion task.

System	Pairwise				B^3		
	Prec.	Rec.	F-0.5	MCC	Prec.	Rec.	F-0.5
Rel-LDA/300	0.593	0.077	0.254	0.191	0.558	0.183	0.396
Rel-LDA/1000	0.638	0.061	0.220	0.177	0.626	0.160	0.396
HAC	0.567	0.152	0.367	0.261	0.523	**0.248**	0.428
Local	0.625	0.136	0.364	0.264	0.626	0.225	0.462
Local+Type	0.718	0.115	0.350	0.265	**0.704**	0.201	0.469
Our Approach	**0.736**	**0.156**	**0.422**	**0.314**	0.677	0.233	**0.490**
Our Approach+Type	0.682	0.110	0.334	0.250	0.687	0.199	0.460

Table 3: Pairwise and B^3 evaluation for various systems. Since our systems predict more fine-grained clusters than Freebase, the recall measure is underestimated.

Path	Example sentence
A beat B	**Dmitry Tursunov** beat the best American player, **Andy Roddick**
A, who lose to B	**Sluman**, Loren Roberts (who lost a 1994 Open playoff to **Ernie Els** at Oakmont ...
A, who beat B	... offender seems to be the Russian **Mariya Sharapova**, who beat **Jelena Dokic**
A, a broker at B	***Robert Bewkes**, a broker at **UBS** for 12 years*
A meet B	**Howell** will meet **Geoff Ogilvy**, Harrington will face Davis Love III

Table 4: A path intrusion task. We show 5 paths and ask the annotator to identify one path which does not belong to the cluster. And we show one example sentence for each path. The entities (As and Bs) in the sentences are bold. And the italic row here indicates the intruder.

System	Correct
Rel-LDA/300	0.737
Rel-LDA/1000	0.821
HAC	0.852
Local+Type	0.773
Our approach	0.887

Table 5: Results of intruding tasks of all systems.

From Table 5, we see that our approach achieves the best performance. We concentrate on some intrusion tasks and compare the clusters produced by different systems.

The clusters produced by HAC (without sense disambiguation) is coherent if all the paths in one relation take a particular sense. For example, one task contains paths “A, director at B”, “A, specialist at B”, “A, researcher at B”, “A, B professor” and “A's program B”. It is easy to identify “A's program B” as an intruder when the annotators realize that the other four paths state the relation that people work in an educational institution. The generative model approach produces more coherent clusters when the number of relation topics increases.

The system which employs local and entity type features (Local+Type) produces clusters with low coherence because the system puts high weight on types. For example, (*United States*, A talk with B, *Syria*) and (*Canada*, A defeat B, *United States*) are clustered into one relation since they share the argument types “country”-“country”. Our approach using the global theme features can correct such errors.

4.3 Error Analysis

We also closely analyze the pairwise errors that we encounter when comparing against Freebase labels. Some errors arise because one instance can have multiple labels, as we explained in Section 4.1. One example is the following: Our approach predicts that (*News Corporation*, buy, *MySpace*) and (*Dow Jones & Company*, the parent of, *The Wall Street Journal*) are in one relation. In Freebase, one is labeled as “/organization/parent/child”, the other is labeled as “/book/newspaper_owner/newspapers_owned”. The latter is a sub-relation of the former. We can overcome this issue by introducing hierarchies in relation labels.

Some errors are caused by selecting the incorrect sense for an entity pair of a path. For instance, we put (*Kenny Smith*, who grew up in, *Queens*) and (*Phil Jackson*, return to, *Los Angeles Lakers*) into the “/people/person/place_of_birth” relation cluster since we do not detect the “sports” sense for the entity pair “(Phil Jackson, Los Angeles Lakers)”.

5 Related Work

There has been considerable interest in unsupervised relation discovery, including clustering approach, generative models and many other approaches.

Our work is closely related to DIRT (Lin and Pantel, 2001). Both DIRT and our approach represent dependency paths using their arguments. Both use distributional similarity to find patterns representing similar semantic relations. Based on DIRT, Pantel et al. (2007) addresses the issue of multiple senses per path by automatically learning admissible argument types where two paths are similar. They cluster arguments to fine-grained entity types and rank the associations of a relation with these entity types to discover selectional preferences. Selectional preferences discovery (Ritter et al., 2010; Seaghdha, 2010) can help path sense disambiguation, however, we show that using global features performs better than entity type features.

Our approach is also related to feature partitioning in cross-cutting model of lexical semantics (Reisinger and Mooney, 2011). And our sense disambiguation model is inspired by this work. There they partition features of words into views and cluster words inside each view. In our case, each sense of a path can be seen as one view. However, we allow different views to be merged since some views overlap with each other.

Hasegawa et al. (2004) cluster pairs of named entities according to the similarity of context words intervening between them. Hachey (2009) uses topic models to perform dimensionality reduction on features when clustering entity pairs into relations. Bollegala et al. (2010) employ co-clustering to find clusters of entity pairs and patterns jointly. All the approaches above neither deal with polysemy nor incorporate global features, such as sentence and document themes.

Open information extraction aims to discover relations independent of specific domains (Banko et al., 2007; Banko and Etzioni, 2008). They employ a self-learner to extract relation instances, but no attempt is made to cluster instances into relations. Yates and Etzioni (2009) present RESOLVER for discovering relational synonyms as a post processing step. Our approach falls into the same category. Moreover, we explore path senses and global features for relation discovery.

Many generative probabilistic models have been applied to relation extraction. For example, varieties of topic models are employed for both open domain (Yao et al., 2011) and in-domain relation discovery (Chen et al., 2011; Rink and Harabagiu, 2011). Our approach employs generative models for path sense disambiguation, which achieves better performance than directly applying generative models to unsupervised relation discovery.

6 Conclusion

We explore senses of paths to discover semantic relations. We employ a topic model to partition entity pairs of a path into different sense clusters and use hierarchical agglomerative clustering to merge senses into semantic relations. Experimental results show our approach discovers precise relation clusters and outperforms a generative model approach and a clustering method which does not address sense disambiguation. We also show that using global features improves the performance of unsupervised relation discovery over using entity type based features.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and the University of Massachusetts gratefully acknowledges the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.

Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16:412–424.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI2007*.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, pages 154–165.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, January.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA. ACM.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of WWW*.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of NIPS*.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. 2011. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of ACL*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 363–370, June.

Benjamin Hachey. 2009. *Towards Generic Relation Extraction*. Ph.D. thesis, University of Edinburgh.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *ACL*.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of KDD*.

J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning Inferential Selectional Preferences. In *Proceedings of NAACL HLT*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of ACL*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*.

Joseph Reisinger and Raymond J. Mooney. 2011. Crosscutting models of lexical semantics. In *Proceedings of EMNLP*.

Bryan Rink and Sanda Harabagiu. 2011. A generative model for unsupervised discovery of relations and argument classes from clinical texts. In *Proceedings of EMNLP*.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A Latent Dirichlet Allocation method for Selectional Preferences. In *Proceedings of ACL10*.

Evan Sandhaus, 2008. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia.

Diarmuid O Seaghdha. 2010. Latent variable models of selectional preference. In *Proceedings of ACL 10*.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of EMNLP*.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34:255–296.