Unsupervised Relation Extraction in Specialized Corpora Using Sequence Mining

Kata Gábor, Haïfa Zargayouna, Isabelle Tellier, Davide Buscaldi, Thierry
Charnois

Unsupervised Relation Extraction in Specialized Corpora using Sequence Mining

Kata Gábor[1], Haïfa Zargayouna[1], Isabelle Tellier[2],
Davide Buscaldi[1], Thierry Charnois[1]

[1] LIPN, CNRS (UMR 7030), Université Paris 13
{firstname.lastname}@lipn.univ-paris13.fr
[2] LaTTiCe, CNRS (UMR 8094), ENS Paris, Université Sorbonne Nouvelle - Paris 3
PSL Research University, Université Sorbonne Paris Cité
isabelle.tellier@univ-paris3.fr

Abstract This paper deals with the extraction of semantic relations from scientific texts. Pattern-based representations are compared to word embeddings in unsupervised clustering experiments, according to their potential to discover new types of semantic relations and recognize their instances. The results indicate that sequential pattern mining can significantly improve pattern-based representations, even in a completely unsupervised setting.

1 Introduction

Relation extraction and classification deal with identifying the semantic relation linking two entities or concepts based on different kinds of information, such as their respective contexts, their co-occurrences in a corpus and their position in an ontology or other kind of semantic hierarchy. It includes the task of finding the instances of the semantic relations, i.e. the entity tuples, and categorizing their relation according to an ontology or a typology. Most systems focus on binary semantic relations. In supervised learning approaches, relation extraction is usually performed in two steps: first, the entity couples corresponding to concepts are extracted or generated, and a binary classification is performed to distinguish those couples which are instances of any semantic relation. Second, the relation itself is categorized according to its similarity to other, known relation types. Unsupervised relation extraction has received far less attention so far. In an unsupervised framework, relation types are inferred directly from the data and, instead of a pre-defined list of relations, new types can be discovered in parallel with the categorization of relation instances.

Unsupervised extraction is often applied to specialized domains, since the manual construction of knowledge bases or training examples for such domains is costly in terms of effort and expertise. The research we present is concerned with unsupervised relation extraction in the scientific domain. The types of relations we expect to extract allow to provide a deeper semantic model and understanding of scientific papers and, in the long term, contribute to automatically build the

state of the art of a given domain from a corpus of articles relevant to it. The deep semantic analysis of scientific literature allows the identification of inter-document links to facilitate inter-textual navigation and the access to semantic context [28]. It also allows to study the evolution of a scientific field over time [6, 26], as well as the access to scientific information through information retrieval [30].

Within this research context, a typology of domain-specific semantic relations was first created and used for corpus annotation in order to confirm the feasibility of the extraction task and to perform extrinsic evaluation of the results [12]. From this point on, our approach to the relation extraction task is completely unsupervised: we do not rely on any of the manually annotated or categorized data. The most important issue consists in defining an approach which is independent from both the domain and the corpus: we do not want to impose any constraint on the types of the relations to be extracted. Different types of information, such as pattern-based representations and word embeddings, are used as input to the classification of the entity couples according to the semantic relation. After performing a range of clustering experiments, we conclude that pattern-based clustering can be significantly improved using sequence mining techniques, yielding the best results in every clustering algorithm we tested.

In what follows, we present the state of the art and explain the specificities of our approach as compared to previous work (2). We then describe the data we used (3), the input features (4) and the algorithms (5). Subsequently, we present the evaluation (6) and discuss the results (6.3). Section 7 concludes the paper and indicates the lines of future work.

2   Related work

Semantic relation extraction and classification is an important task in the domain of information extraction, knowledge extraction and knowledge base population. A plethora of approaches have been applied to relation extraction, among which we can distinguish tendencies according to:

- whether the method aims to classify entity couples in a given set of relations or to discover new types of relations,
- the approach to be used: symbolic approaches through e.g. hand-crafted extraction patterns, or machine learning approaches (classification/clustering or sequence labeling methods),
- the input information and the representation used: pattern-based, lexical, syntactic features, distributional vectors, etc.

Supervised systems rely on a list of pre-defined relations and categorized examples, as described in the shared tasks of MUC and ACE campaigns [15]. Using small, manually annotated training corpora, these systems extract different kinds of features eventually combined with external knowledge sources, and build classifiers to categorize new relationship mentions [38]. Symbolic systems, similarly to supervised learning algorithms, are specific to the list of relations they are designed to recognize - based on hand-crafted linguistic rules created

by linguists or domain experts [13]. On the other hand, with the proliferation of available corpora, a new task emerged: *Open Information Extraction (OpenIE)* [1, 8]. It aims at developing unsupervised or distantly supervised systems with a double objective: overcome the need of scarcely available annotated data, and ensure domain-independence by being able to categorize instances of new relation types. In this kind of work, applications are not limited to a given set of relations and become able to cope with the variety of domain-specific relations [1,10]. Such experiments can also be beneficial for the automated population of ontologies [27] or thesauri [10]. Our work belongs to that second line of research.

According to the type of input features which serve as a base for classification, we can distinguish pattern-based approaches from classification approaches relying on diverse quantifiable attributes. The hypothesis behind pattern-based approaches is that the semantic relation between two entities is explicit in at least a part of their co-occurrences in the text, and therefore relation instances can be identified based on text sequences between/around the entities. Such characteristic patterns are usually manually defined, incorporating linguistic and/or domain knowledge in rule-based approaches [13, 34]. Patterns are not limited to sequences of words, they can contain a combination of lexical and syntactic information [9]. Patterns can also be used indirectly as inputs to supervised classifiers [1] or for calculating similarities between entities' distribution over patterns [20, 32]. Most of these approaches rely on hand-crafted lists of patterns. In [4], sequential pattern mining is used to discover new linguistic patterns within the framework of a symbolic approach.
Another way of including quantifiable context features for relation extraction is to use distributional word vectors, either as "count models" [3] or as word embeddings [21]. Entity couples can be represented by a vector built from the vectors associated with each of its members: popular methods include concatenating the two vectors [2] or taking their difference [33]. These representations will then serve as input for a supervised classifier. However, it has recently been argued in [19] that both concatenation and difference are "clearly ignoring the relation between x and y" (i.e. what links the entities): they only provide information on the type of the individual entities. In this article, the conclusion was that "contextual features might lack the necessary information to deduce how one word relates to another".

Finally, certain biclustering or iterative clustering methods are sometimes applied to divide not only the objects (word or entity couples), but also the dimensions (patterns or features) in parallel. Generative models are more prevalent in this framework. In [35] Latent Dirichlet Allocation (LDA) is adapted to the task of unsupervised relation classification. In [16] Markov logic is used, while in [24] an iterative soft clustering algorithm is applied, based on a combination of distributional similarities and a heuristic method for mining hypernyms in the corpus.

The approach we put forward belongs to the unsupervised/OpenIE framework. We do not rely on any manually classified data or typology of relations. Our

experiments rely on unsupervised clustering using two types of representations: text patterns and word embeddings. Moreover, we make use of sequential pattern mining in order to enrich our couples of entities/text patterns matrix and address data sparsity. Our experiments were conducted on the ACL Anthology Corpus of computational linguistics papers, but they can be applied to any field in the scientific domain. In the context of our work, the final purpose is to extract the state of the art of a scientific domain, therefore the constitution of the corpus and the evaluations are focused on the relation types relevant for this kind of information; however, this context does not directly influence our choice of representation and clustering algorithm. Our approach differs from standard relation classification tasks, as defined e.g. in SemEval campaigns [14] in two respects. First, we do not target relations belonging to a pre-defined set. Second, the semantic relations considered in SemEval were lexical by nature, e.g.:

Component-Whole *Example: My apartment has a large kitchen.*
Member-Collection *Example: There are many trees in the forest.*

On the contrary, the relations we hope to extract are largely contextual. The same couple of entities can instantiate several distinct relations in the same corpus in different contexts:

Uses_information: (...) **models** extract rules using **parse trees** (...)
Used_for: (...) **models** derive **discourse parse trees** (...)

3   Data and Resources

For the purpose of these experiments, we used a corpus where concepts in the scientific domain are annotated. The corpus is extracted from the ACL Anthology Corpus [29]. We decided to focus on the abstract and introduction parts of scientific papers since they express essential information in a compact and often repetitive manner, which makes them an optimal source for mining sequential patterns. The resulting corpus of abstracts and introductions contains 4,200,000 words from 11,000 papers.

Entity annotation was done in two steps. First, candidates were generated with the terminology extraction tool TermSuite [7]. The list of extracted terms was then mapped to different ontological resources: the knowledge base of Saffron Knowledge Extraction Framework [5], and the BabelNet ontology [25]. If a term was validated as a domain concept (i.e., found in at least one of the resources), it was annotated in the text. The reader is referred to [11] for further information on the annotated corpus.

4   Input representations

The goal of this part is to represent each co-occurring entity couple in a vector space which allows to calculate a similarity between them. Three distinct types of vector spaces were used as representation bases for our clustering experiments. The first two are pattern-based: they rely on the assumption that couples of

entities linked by the same semantic relation will be characterized by similar patterns in at least a part of their co-occurrence contexts (i.e. in the text between the two elements of the couple). One of the representations uses complete text sequences as they are found in the corpus, while the other one relies on *patterns* that were extracted from these sequences using sequential pattern mining.

The expected advantages of identifying patterns inside the sequences are similar to those using distributed representations. First, using the complete sequences as features leads to data sparsity. Although patterns are basically sub-sequences of the sequences in the string representation and thus we can expect the size of the feature space to grow, the same sequence can belong to more than one pattern, and thus the number of *frequent* features is also expected to grow. Second, while adding some words to a sequence may not modify its meaning and the relationship between the two entities, it will still result in separate features in the full sequence representation. A pattern-based representation can capture and quantify the elements of similarity between close, but not identical sequences. Finally, sub-sequences can encode different types of information, e.g. grammatical words can be relevant for the relation between the entities, while content words will provide information about the topic of the context, and both kinds of information are expected to bring us closer to characterizing the semantic relation.

The third representation uses word embeddings of the entities considered separately and hypotheses that their semantic relation is mainly context-independent. By calculating the pairwise similarity between the entities, we expect to quantify the similarity between relation instances. This representation is similar to the one used in [2], though the scope of the experiments and the classification method are different.

4.1 Pattern-based representations

In the pattern-based representation, attributes correspond to text sequences that are found between co-occurring entity couples. We extracted from our corpus every entity couple occurring in the same sentence, together with the text between them. Text sequences can contain other entities, but their length is limited to ≤ 8 words. This results in 998,000 instances extracted.

String representation Using these co-occurrence data, we first built a sparse matrix M with lines corresponding to entity couples $e=(e_1, e_2)$ and columns corresponding to text sequences $p \in P$. The cells $M_{e,p}$ contain an association value between e et p. One of the representations uses raw co-occurrence count, while the other one uses $PPMI_\alpha$ weighting. This weight is a variant of Pointwise Mutual Information (PMI) in which values below 0 are replaced by 0. Moreover, the context distribution smoothing method proposed by [18] is applied to the positive PMI weighting. This smoothing, inspired by the success of neural word embeddings in semantic similarity tasks [3, 21], allows to reduce the bias of PMI towards rare contexts. Context words' co-occurrence counts are raised to the

power of α (in equation (2)). Its optimal value is reported to be $\alpha = 0.75$ according to the experiments of [22]. This finding was directly adapted to PMI [18]:

$$PPMI_\alpha(e,p) = max(log_2 \frac{P(e,p)}{P(e) \times P_\alpha(p)}, 0) \qquad (1)$$

$$P_\alpha(p) = \frac{freq(c)^\alpha}{\sum_c freq(c)^\alpha} \qquad (2)$$

We will refer to the vectors built as such as the **string** representation.

Sequential pattern representation For the second experiment, we applied sequential pattern mining techniques [31] to discover relevant patterns which are specific to semantic relations. The extraction is completely unsupervised: frequent sequential patterns which fulfill a certain number of constraints are automatically extracted from the input. A sequence, in this context, is a list of literals (items) and an item is a word in the corpus. The input corpus was made of all the sequences extracted from co-occurring entities (i.e. the feature space for the **string** representation). The pattern mining process is applied to word forms without using any additional linguistic information.

The sequence mining tool [4] we used allows distinct options to add constraints on the extracted sequences. We selected contiguous sequences of length between 2 and 8 words and a minimum support of 10. The support of a sequential pattern in a sequence database is the number of sequences in the database containing the pattern. Only *closed* sequential patterns were considered, i.e. patterns which are not sub-sequences of another sequence with the same support.

To construct the matrices, we filled the cells with raw co-occurrences (how many times a pattern occurs somewhere between the two entities) and, for a second matrix, with the PPMI$_\alpha$-weighted values. We will refer to this representation as **pattern** representation.

4.2   Distributional representation

This type of feature space also uses contextual information, but it is computed independently for the two entities. We used word2vec [21] to create the distributional vectors, as it proved to be particularly well adapted for semantic similarity tasks and is presumed to encode analogies between semantic relations [23]. word2vec was trained on the whole ACL Anthology Corpus using the skip-gram model [21] and the resulting word embeddings (size=200) were used to represent each entity.

The vector of an entity couple is simply made of the concatenation of the vectors of each entity[3] [2]. We expect this representation to capture very specific relation types, where the potential arguments belong to a restricted semantic class.

---

[3] Entities corresponding to multiword expressions will have their unique vector, since word2vec includes an internal module for recognizing multiword expressions.

5   Clustering

Two methods of hierarchical clustering were tested using cosine similarity and Cluto's [37] clustering functions. The first one is a top-down clustering based on repeated bisections: at every iteration, a cluster is split in two until the desired number of clusters is reached. This number has to be pre-defined: experiments were performed using different values. The cluster to be divided is chosen so that it maximizes the sum of inter-cluster similarities for each resulting cluster. We will refer to this method as **divisive**.

The second method is a hierarchical agglomerative clustering with a bisective initialization [36] : a clustering in $\sqrt{n}$ clusters is first calculated (where n is the number of clusters to be produced) through repeated bisections. The vector space is then augmented with $\sqrt{n}$ new dimensions that represent the clusters calculated at the first step, and the values of these dimensions are given by the distance of each object from the centroids of the clusters produced at the initiation stage. The agglomerative clustering is then performed on this augmented vector space. This method was created to combine advantages of divisive (global) and agglomerative (local) algorithms by reducing the impact of errors from initial merging decisions, which tend to be multiplied as the agglomeration progresses [36].

We will refer to this algorithm as **agglo**.

6   Evaluation

6.1   Standard classification

For the sake of the experiments, we selected a sample of 500 abstracts (about 100 words/abstract) and manually annotated relevant relations occurring in this sample. The typology of relations was data-driven: it was established in parallel with the categorization of the examples. An illustration of the relations we identified is shown in Table 1, for a complete description of the manual annotation work and the typology, see [12]. The relations are not specific to the natural language processing domain; they can be used for any scientific corpora.

char	ARG1: *observed characteristics* of an observed ARG2: *entity*
composed_of	ARG1: *database/resource* ARG2: *data*
methodapplied	ARG1: *method* applied to ARG2: *data*
model	ARG1: *abstract representation* of an ARG2: *observed entity*
phenomenon	ARG1: *entity, a phenomenon* found in ARG2: *context*
propose	ARG1: *paper/author* presents ARG2: *an idea*

Table 1. Extract of the typology of semantic relations

As a second step, a sample of 615 entity couples which co-occur in the corpus was manually categorized according to this typology. This sample was used as a gold standard for clustering evaluation.

6.2   Baseline and evaluation measures

The clustering results were compared to the standard one as a series of decisions: whether to classify two couples in the same group or in different groups. This evaluation is less influenced by structural differences between two clustering solutions and allows to quantify results in terms of precision and recall.
We also calculated APP *Adjusted Pairwise Precision* [17]: this measure quantifies average cluster purity, weighted by the size of the clusters. This provides additional information on the proportion of the relevant clusters.

$$APP = \frac{1}{|K|} \sum_{i=1}^{|K|} \frac{nb\ correct\ pairs\ in\ k_i}{nb\ pairs\ in\ k_i} \times \frac{|k_i| - 1}{|k_i| + 1} \qquad (3)$$

For each experiment with respect to cluster size, we created a corresponding random clustering to estimate the difficulty of the task and the contribution of our approaches.

6.3   Results and discussion

The evaluation was conducted so as to allow comparisons between the two clustering algorithms, the three input representations and the two weighting systems. Cluster sizes have an important effect on the results because they are correlated with the number of classes in the standard (21 in our case). On the other hand, a different cluster structure e.g. with finer grained distinctions, may also be semantically justified. The real validity of the clusters must therefore be established by human inspection.

Input	#clusters	algorithm	weight	APP	Prec	Recall	F-measure
baseline	100	random	N/A	0.0813	0.0955	0.0097	0.0176
baseline	50	random	N/A	0.0883	0.1036	0.0198	0.0332
baseline	25	random	N/A	**0.0979**	**0.1040**	**0.0410**	**0.0588**
string	100	divisive	freq	0.2498	0.3037	0.1030	0.1538
pattern	100	divisive	freq	0.2823	**0.3718**	0.0993	0.1568
string	50	divisive	freq	0.2985	0.2805	0.1302	0.1778
pattern	50	divisive	freq	0.3265	0.3159	0.1235	0.1776
string	25	divisive	freq	0.3941	0.2219	**0.1904**	0.2050
pattern	25	divisive	freq	**0.3947**	0.2776	0.1773	**0.2164**
word2vec	100	divisive	incl	0.3396	**0.5734**	0.0527	0.0965
word2vec	50	divisive	incl	0.3541	0.4761	0.0890	0.1499
word2vec	25	divisive	incl	**0.3545**	0.4182	**0.1539**	**0.2250**

Table 2. Clustering results with the divisive algorithm

Table 2 shows the results of the **divisive** method. The **string** and **pattern** representations with raw frequency counts are compared with the baseline and

word2vec vector representations (where weights are implicitly included in the language model learning). Although the word2vec representation yields a very good performance with respect to both precision measures, it comes at the cost of a very low recall. Since this representation is solely based on the similarity between individual entities, this means that mainly couples having nearly identical entities end up in the same cluster, e.g. : *parsing - sentences, parses - sentences, parse - sentence.* In agreement with [19], this result reveals that this representation is not good at capturing relational similarities.

Input	#clusters	algorithm	weight	APP	Prec	Recall	F-measure
string	100	divisive	$PPMI_\alpha$	0.3112	**0.4905**	0.0462	0.0844
string	50	divisive	$PPMI_\alpha$	**0.3625**	0.3789	0.0799	0.1320
string	25	divisive	$PPMI_\alpha$	0.3555	0.3133	**0.1400**	**0.1936**

Table 3. The effect of the $PPMI_\alpha$ weighting with the divisive algorithm

Another interest of Table 2 is the improvement in precision (for both measures) brought by the **pattern** representation, compared to the **string** representation. This improvement is accompanied by slight decreases in recall. It is also interesting to note that, as shown by Table 3, $PPMI_\alpha$ weighting transforms sequence-based scores the same way as to what we observe with **word2vec** representations: very high precision with very low recall -despite the fact that the semantics captured by the input representations are different in both cases.

Input	#clusters	algorithm	weight	APP	Prec	Recall	F-measure
string	100	agglo	$PPMI_\alpha$	**0.3020**	0.4184	0.1582	0.2296
pattern	100	agglo	$PPMI_\alpha$	0.2810	**0.4758**	0.1936	0.2752
string	50	agglo	$PPMI_\alpha$	0.2535	0.3246	0.2142	0.2581
pattern	50	agglo	$PPMI_\alpha$	0.2697	0.4200*	0.2657*	0.3268
string	25	agglo	$PPMI_\alpha$	0.2585	0.2898	0.2277	0.2550
pattern	25	agglo	$PPMI_\alpha$	0.2460	0.3777*	**0.2914**	**0.3290**
word2vec	100	agglo	incl	**0.3630**	**0.5285**	0.1316	0.2107
word2vec	50	agglo	incl	0.2966	0.3694	0.1938	0.2542
word2vec	25	agglo	incl	0.2972	0.3330	**0.2399**	**0.2789**

Table 4. Clustering results with the agglomerative algorithm

Table 4 presents the results of the **agglo** clustering method. This algorithm works better for every type of representation we considered. The scores reported here are obtained on $PPMI_\alpha$-weighted string and pattern representations. The **pattern** representation comes out as the absolute winner, with important im-

provements over **string** both in terms of precision (6-10%) and recall (3.5-6.5%). Scores marked by * indicate statistically significant improvements according to a 10-fold cross-validation on the **string** and **pattern** clustering solutions. The **pattern** representation also beats the precision of **word2vec** in two out of the three settings. Although the recall obtained with the word2vec vectors is also improved by the agglomerative method, the **pattern** representation still holds an important advantage.

7   Conclusion and future work

We presented an approach to extract new types of semantic relations and instances of relations from specialized corpora using unsupervised clustering. Two types of representations were compared: pattern-based vectors and word embeddings. In agreement with previous results, we found that concatenated word embeddings tend to have a limited contribution to discovering new relation types.

An important finding is that sequential pattern mining contributes to create a much more adapted feature space, as shown by the significant improvement both in terms of precision and recall. This confirms our expectation that sequential patterns are better than full sequences in capturing relational similarities. Another advantage is that the pattern mining process is completely unsupervised.
We plan to conduct a manual evaluation of the resulting clusters. This would allow to have a better insight on the nature of the resulting clusters. Biclustering methods can also be tested on the data: they have the potential to automatically identify the most relevant patterns for each relation type.

References

1. M. Banko, J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.
2. M. Baroni, R. Bernardi, N-Q. Do, and C-C. Shan. Entailment above the word level in distributional semantics. In *ACL '12*, 2012.
3. M. Baroni, G. Dinu, and G. Kruszewski. Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL '14*, 2014.
4. N. Béchet, P. Cellier, T. Charnois, and B. Crémilleux. Discovering linguistic patterns using sequence mining. In *CICLing '12*, 2012.
5. G. Bordea, P. Buitelaar, and T. Polajnar. Domain-independent term extraction through domain modelling. In *TIA '13*, 2013.
6. D. Chavalarias and J-P. Cointet. Phylomemetic patterns in science evolution - the rise and fall of scientific fields. *PLOS ONE*, 8(2), 2013.

7. B Daille. Building bilingual terminologies from comparable corpora: The ttc termsuite. In *5th Workshop on Building and Using Comparable Corpora, co-located with LREC*, pages 39–32, 2012.
8. L. Del Corro and R. Gemulla. Clausie: Clause-based open information extraction. In *International Conference on World Wide Web*, WWW '13, 2013.
9. A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP '11*, 2011.
10. O. Ferret. *Language Production, Cognition, and the Lexicon*, chapter Typing Relations in Distributional Thesauri, pages 113–134. Springer International Publishing, 2015.
11. K. Gábor, H. Zargayouna, D. Buscaldi, I. Tellier, and T. Charnois. Semantic annotation of the acl anthology corpus for the automatic analysis of scientific literature. In *LREC '16*, Portoroz, Slovenia, 2016. in press.
12. K. Gábor, H. Zargayouna, I. Tellier, D. Buscaldi, and T. Charnois. A typology of semantic relations dedicated to scientific literature analysis. In *SAVE-SD Workshop at the 25th World Wide Web Conference*, 2016.
13. M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING '92*, page 539–545, 1992.
14. I. Hendrickx, S. N. Kim, Z. Kozareva, D. Nakov, P.and O Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations*, 2010.
15. Jerry R. Hobbs and Ellen Riloff. Information extraction. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010.
16. S. Kok and P. Domingos. Extracting semantic networks from text via relational clustering. In *Proceedings of ECML PKDD'08*, 2008.
17. A. Korhonen, Y. Krymolowski, and N. Collier. The choice of features for classification of verbs in biomedical texts. In *COLING*, 2008.
18. O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the ACL*, 3, 2015.
19. O. Levy, S. Remus, Biemannm C., and I. Dagan. Do supervised distributional methods really learn lexical inference relations? In *ACL '15*, 2015.
20. D. Lin and P. Pantel. Dirt: Discovery of inference rules from text. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2001.
21. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.
22. T. Mikolov, I. Sutskever, K. Chen, GS. Corrado, and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
23. T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *NAACL*, 2013.
24. B. Min, S. Shi, R. Grishman, and C.-Y. Lin. Ensemble semantics for large-scale unsupervised relation extraction. In *EMNLP'12*, 2012.
25. R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193, 2012.
26. E. Omodei, J-P. Cointet, and T. Poibeau. Mapping the natural language processing domain : Experiments using the acl anthology. In *LREC '14*, 2014.

27. G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos. Ontology population and enrichment: State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution.* Springer-Verlag, 2011.

28. Valentina Presutti, Sergio Consoli, Andrea Giovanni Nuzzolese, Diego Reforgiato Recupero, Aldo Gangemi, Ines Bannour, and Haïfa Zargayouna. Uncovering the semantics of wikipedia pagelinks. In *Knowledge Engineering and Knowledge Management*, pages 413–428. Springer, 2014.

29. D.R. Radev, P. Muthukrishnan, and V. Qazvinian. The ACL Anthology Network Corpus. In *ACL Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, 2009.

30. Bahar Sateli and René Witte. What's in this paper?: Combining rhetorical entities with linked open data for semantic literature querying. In *Proceedings of the 24th International Conference on World Wide Web*, 2015.

31. Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, pages 3–17, 1996.

32. P. D. Turney. Similarity of semantic relations. *CoRR*, abs/cs/0608100, 2006.

33. J. Weeds, D. Clarke, J. Reffin, D. Weir, and B. Keller. Learning to distinguish hypernyms and co-hyponyms. In *COLING '14*, 2014.

34. R. Yangarber, W. Lin, and R. Grishman. Unsupervised learning of generalized names. In *COLING '02*, 2002.

35. L. Yao, A. Haghighi, S. Riedel, and A. McCallum. Structured relation discovery using generative models. In *EMNLP'11*, 2011.

36. Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM*, 2002.

37. Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining for Knowledge Discovery*, 10, March 2005.

38. G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *ACL '05*, 2005.