Relation extraction pattern ranking using word similarity

Konstantinos Lambrou-Latreille
(1) École Polytechnique de Montréal
(2) Centre de Recherche Informatique de Montréal
Montréal, Québec, Canada
konstantinos.lambrou-latreille@polymtl.ca

Abstract

Our thesis proposal aims at integrating word similarity measures in pattern ranking for relation extraction bootstrapping algorithms. We note that although many contributions have been done on pattern ranking schemas, few explored the use of word-level semantic similarity. Our hypothesis is that word similarity would allow better pattern comparison and better pattern ranking, resulting in less semantic drift commonly problematic in bootstrapping algorithms. In this paper, as a first step into this research, we explore different pattern representations, various existing pattern ranking approaches and some word similarity measures. We also present a methodology and evaluation approach to test our hypothesis.

1 Introduction

In this thesis, we look at the problem of information extraction from the web; more precisely at the problem of extracting structured information, in the form of triples (predicate, subject, object), e.g. (ObjectMadeFromMaterial, table, wood) from unstructured text. This topic of Relation Extraction (RE), is a current and popular research topic within NLP, given the large amount of unstructured text on the WWW.

In the literature, machine learning algorithms have shown to be very useful for RE from textual resources. Although supervised (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005) and unsupervised learning (Hasegawa et al., 2004; Zhang et al., 2005) have been used for RE, in this thesis, we will focus on semi-supervised bootstrapping algorithms.

In such algorithms (Brin, 1999; Agichtein and Gravano, 2000; Alfonseca et al., 2006a), the input is a set of related pairs called *seed instances* (e.g., (table,wood), (bottle, glass)) for a specific *relation* (e.g., ObjectMadeFromMaterial). These seed instances are used to collect a set of *candidate patterns* representing the relation in a corpus. A subset containing the best candidate patterns is added in the set of *promoted patterns*. The promoted patterns are used to collect *candidate instances*. A subset containing the best candidate instances is selected to form the set of *promoted instances*. The promoted instances are either added to the initial seed set or used to replace it. With the new seed set, the algorithm is repeated until a stopping criterion is met.

The advantage of bootstrapping algorithms is that they require little human annotation. Unfortunately, the system may introduce wrongly extracted instances. Due to its iterative approach, errors can quickly cumulate in the next few iterations; therefore, precision will suffer. This problem is called *semantic drift*. Different researchers have studied how to counter *semantic drift* by using better pattern representations, by filtering unreliable patterns, and filtering wrongly extracted instances (Brin, 1999; Agichtein and Gravano, 2000; Alfonseca et al., 2006a). Nevertheless, this challenge is far from being resolved, and we hope to make a contribution in that direction.

The semantic drift is directly related to which candidate patterns become promoted patterns. A crucial decision at that point is how to establish pattern confidence so as to rank the patterns. There are many ways to estimate the confidence of a pattern.

Proceedings of NAACL-HLT 2015 Student Research Workshop (SRW), pages 25–32,
Denver, Colorado, June 1, 2015. ©2015 Association for Computational Linguistics

Blohm et al. (2007) identified general types of pattern filtering functions for well-known systems. As we review pattern ranking approaches, we note that many include a notion of ”resemblance”, as either comparing patterns between successive iterations, or comparing instances generated at an iteration to instances in the seed set, etc. Although this notion of resemblance seems important to many ranking schemas, we do not find much research which combines word similarity approaches within pattern ranking. This is where we hope to make a research contribution and where our hypothesis lies, that using word similarity would allow for better pattern ranking.

In order to suggest better pattern ranking approaches incorporating word similarity, we need to look at the different pattern representations suggested in the literature and understand how they affect pattern similarity measures. This is introduced in Section 2. Then, section 3 provides a non-exhaustive survey of pattern ranking approaches with an analysis of commonality and differences; Section 4 presents a few word similarity approaches; Section 5 presents the challenges we face, as well as our methodology toward the validation of our hypothesis; Section 6 briefly explores other anticipated issues (e.g. seed selection) in relation to our main contribution and Section 7 presents the conclusion.

2 Pattern representation

In the literature, pattern representations are classified as lexical or syntactic.

Lexical patterns represent lexical terms around a relation instance as a pattern. For relation instance (*X,Y*) where *X* and *Y* are valid noun phrases, Brin (1999), Agichtein and Gravano (2000), Pasca et al. (2006), Alfonseca et al. (2006a) take *N* words before *X*, *N* words after *Y* and all intervening words between *X* and *Y* to form a pattern (e.g., *well-known author X worked on Y daily.*). Extremes for the choice of N exist, as in the CPL subsystem of NELL (Carlson et al., 2010) setting $N = 0$ and the opposite in Espresso (Pantel and Pennacchiotti, 2006) where the whole sentence is used.

Syntactic patterns convert a sentence containing a relation instance to a structured form such as a parse tree or a dependency tree. Yangarber (2003) and Stevenson and Greenwood (2005) use Subject-Verb-Object (SVO) dependency tree patterns such as [Company appoint Person] or [Person quit]. Culotta (2004) uses full dependency trees on which a tree kernel will be used to measure similarity. Bunescu and Mooney (2005) and Sun and Grishman (2010) use the shortest dependency path (SDP) between a relation instance in the dependency tree as a pattern (e.g., ”nsubj ← met → prep_in”). Zhang et al. (2014) add a semantic constraint to the SDP; they define the semantic shortest dependency path (SSDP) as a SDP containing at least one *trigger word* representing the relation, if any. Trigger words are defined as words most representative of the target relation (e.g. *home*, *house*, *live*, for the relation *PersonResidesIn*).

We anticipate the use of word similarity to be possible when comparing either lexical or syntactic patterns, adapting to either words in sequence, or nodes within parse or dependency trees. In fact, as researchers have explored pattern generalization, some have already looked at ways of grouping similar words. For example, Alfonseca et al. (2006a) present a simple algorithm to generalize the set of lexical patterns using an edit-distance similarity. Also, Pasca et al. (2006) add term generalization to a pattern representation similar to Agichtein and Gravano (2000); terms are replaced with their corresponding classes of distributionally similar words, if any (e.g., let CL3 = {March, October, April,...} in the pattern *CL3 00th : X’s Birthday (Y)*).

3 Pattern ranking approaches

We now survey pattern ranking algorithms to better understand in which ones similarity measures would be more likely to have an impact. We follow a categorization introduced in Blohm et al. (2007) as they quantified the impact of different relation pattern/instance filtering functions on their generic bootstrapping algorithm. The filtering functions proposed by Brin (1999), Agichtein and Gravano (2000), Pantel and Pennacchiotti (2006) and Etzioni et al. (2004) were described in their work.

Although non-exhaustive, our survey includes further pattern ranking approaches found in the literature, in order to best illustrate Blohm’s different categories. A potential use of those categories would be to define a pattern ranking measure composed of voting experts representing each category. A combination of these votes might provide a better confidence measure for a pattern.

We define the following notation, as to allow the description of the different measures in a coherent way. Let p be a pattern and i be an instance; I is the set of promoted instances; P is the set of promoted patterns; $H(p)$ is the set of unique instances matched by p; $K(i)$ is the set of unique patterns matching i; $count(i,p)$ is the number of times p matches i; $count(p)$ is the number of p occurs in a corpus; S is the set of seed instances.

3.1 Syntactic assessment

This filtering assessment is purely based on the syntactic criteria (e.g., length, structure, etc.) of the pattern. Brin (1999) uses the length of the pattern to measure its specificity.

3.2 Pattern comparison

Blohm et al. (2007) named this category *inter-pattern comparison*. Their intuition was that candidate patterns could be rated based on how similar their generated instances are in comparison to the instances generated by the promoted patterns. We generalize this category to also include rating of candidate patterns based directly on their semantic similarity with promoted pattern.

Stevenson and Greenwood (2005) assign a score on a candidate pattern based on the similarity with promoted patterns. The pattern scoring function uses the Jiang and Conrath (1997) WordNet-based word similarity for pattern similarity. They represent the SVO pattern as a vector (e.g., [subject_COMPANY, verb_fired, object_ceo], or [subject_chairman, verb_resign]). The similarity between two pattern vectors is measured as :

$$sim(\vec{a},\vec{b}) = \frac{\vec{a} \times W \times \vec{b^T}}{|\vec{a}| \times |\vec{b}|} \quad (1)$$

where W is a matrix that contains the word similarity between every possible element-filler pairs (e.g., subject_COMPANY, verb_fired, object_ceo) contained in every SVO pattern extracted from a corpus. The top-*N* (e.g., 4) patterns with a score larger than 95% are promoted.

Zhang et al. (2014) defines a bottom-up kernel (BUK) to filter undesired relation patterns. The BUK measures the similarity between two dependency tree patterns. The system accepts new patterns that are the most similar to seed patterns. The BUK defines a matching function t and a similarity function k on dependency trees. Let dep be the pair (rel, w) where rel is the dependency relation and w is the word of the relation (e.g., (nsubj, son)). The matching function is defined as:

$$t(dep_1, dep_2) = \begin{cases} 1 & if\ dep_1.w, dep_2.w \in W_{tr} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where W_{tr} is the set of trigger words for the target relation. The similarity function is defined as:

$$k(dep_1, dep_2) = \begin{cases} \gamma_1 + \gamma_2 & if\ dep_1.rel = dep_2.rel\ \&\&\ dep_1.w = dep_2.w \\ \gamma_1 & if\ dep_1.w = dep_2.w \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where γ_1 and γ_2 are manually defined weights for attributes $dep.w$ and $dep.rel$ respectively. The word comparison is string-based.

3.3 Support-based assessment

This ranking assessment estimates the quality of a pattern based on the set of occurrences/patterns that generated this pattern. This assessment is usually used for patterns that were created by a generalization procedure. For example, if pattern *X BE mostly/usually made of/from Y* was generated by patterns *X is usually made of Y* and *X are mostly made from Y*, then the quality of the generalized pattern will be based on the last two patterns. Brin (1999) filters patterns if $(specificity(p) \times n) > t$, where *n* is the occurrence count of pattern *p* applied in a corpus and *t* is a manually set threshold.

3.4 Performance-based assessment

The quality of a candidate pattern can be estimated by the comparing its correctly produced instances with the set of promoted instances.

Blohm et al. (2007) defines a precision formula similar to Agichtein and Gravano (2000) to approximate a performance-based precision:

$$prec(p) = \frac{|H(p) \cap S|}{|H(p)|} \quad (4)$$

Alfonseca et al. (2006b) propose a procedure to measure the precision of candidate patterns in order to filter overly-general patterns. For every relation, and every hook *X* and target *Y* of the set of promoted instances (*X,Y*), a hook and target corpus is extracted from corpus *C*; *C* contains only sentences which contain *X* or *Y*. For every pattern p, instances of $H(p)$ are extracted. Then, a set of heuristics label every instance as correct/incorrect. The precision of p is number of correct extracted instances divided by the total number of extracted instances.

NELL (Carlson et al., 2010) ranks relation patterns by their precision:

$$prec(p) = \frac{\sum_{i \in I} count(i,p)}{count(p)} \tag{5}$$

Sijia et al. (2013) filters noisy candidate relation patterns that generate instances which appear in the seed set of relations other than the target relation.

3.5 Instance-Pattern correlation

Pattern quality can be assessed by measuring its correlation with the set of promoted instances. These measures estimate the correlation by counting pattern occurrences, promoted instance occurrences, and pattern occurrences with a specific promoted instance.

Blohm et al. (2007) classified Espresso (Pantel and Pennacchiotti, 2006) and KnowItAll (Etzioni et al., 2004) in this category.

Pantel et Pennacchiotti (2006) ranks candidate relation patterns by the following reliability score:

$$r_\pi(p) = \frac{\sum_{i \in I} \left(\frac{pmi(i,p)}{max_{pmi}} \times r_l(i) \right)}{|I|} \tag{6}$$

where max_{pmi} is the maximum PMI between all pattern and all instances, and $pmi(i,p)$ can be estimated using the following formula:

$$pmi(i,p) = \log \left(\frac{|x,p,y|}{|x,*,y| \times |*,p,*|} \right) \tag{7}$$

where i is an instance (*x,y*), $|x,p,y|$ is the occurrence of pattern *p* with terms *x* and *y* and (*) represents a wildcard. The reliability of an instance $r_l(i)$ is defined as:

$$r_l(i) = \frac{\sum_{p \in P} \left(\frac{pmi(i,p)}{max_{pmi}} \times r_\pi(p) \right)}{|P|} \tag{8}$$

Since $r_l(i)$ and $r_\pi(p)$ are defined recursively, $r_l(i) = 1$ for any seed instance. The top-*N* patterns are promoted where *N* is the number of patterns of the previous bootstrapping iteration plus one.

Sun and Grishman (2010) accept the top-*N* ranked candidate pattern by the following confidence formula:

$$Conf(p) = \frac{Sup(p)}{|H(p)|} \times \log Sup(p) \tag{9}$$

where $Sup(p) = \sum_{i \in H(p)} Conf(i)$ is the support candidate pattern *p* can get from the set of matched instances. Every relation instance in Sun and Grishman (2010) has a cluster membership, where a cluster contains similar patterns. The confidence of an newly extracted instance i is defined as:

$$Conf(i) = 2 \times \frac{Semi\_Conf(i) \times Cluster\_Conf(i)}{Semi\_Conf(i) + Cluster\_Conf(i)} \tag{10}$$

$$Semi\_Conf(i) = 1 - \prod_{p \in K(p)} (1 - Prec(p)) \tag{11}$$

$$\begin{aligned} Cluster\_Conf(i) &= Prob(i \in C_t) \\ &= \frac{\sum_{p \in C_t} count(i,p)}{|K(i)|} \end{aligned} \tag{12}$$

where C_t is the target cluster where the patterns of the target relation belong, $Semi\_Conf(i)$ is defined as the confidence given by the patterns matching the candidate relation instance and $Cluster\_Conf(i)$ is defined how strongly a candidate instance is associated with the target cluster.

4 Word similarity

Within the pattern ranking survey, we often saw the idea of comparing patterns and/or instances, but only once, was there a direct use of word similarity measures. Stevenson and Greenwood (2005) assign a score to a candidate pattern based on its similarity to promoted patterns using a WordNet-based word similarity measure (Jiang and Conrath, 1997). This measure is only one among many WordNet-based approaches, as can be found in (Lesk, 1986; Wu and Palmer, 1994; Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998; Leacock and Chodorow, 1998; Banerjee and Pedersen, 2002).

There are limitations to these approaches, mainly that WordNet (Miller, 1995), although large, is still incomplete. Other similarity approaches are corpus-based (e.g. (Agirre et al., 2009)) where the distributional similarity between words is measured. Words are no longer primitives, but they are represented by a feature vector. The feature vector could contain the co-occurrences, the syntactic dependencies, etc. of the word with their corresponding frequencies from a corpus. The cosine similarity (among many possible measures) between the feature vector of two words indicates their semantic similarity.

Newer approaches to word similarity are based on neural network word embeddings. Mikolov et al. (2013) present algorithms to learn those distributed word representations which can then be compared to provide word similarity estimations.

Word similarity could be in itself the topic of a thesis. Therefore, we will not attempt at developing new word similarity measures, but rather we will search for measures which are intrinsically good and valuable for the pattern ranking task. The few mentioned above are a good start toward a more extensive survey. The methods found can be evaluated on existing datasets such as RG (Rubenstein and Goodenough, 1965), MC (Miller and Charles, 1991), WordSim353 (Finkelstein et al., 2001; Agirre et al., 2009), MTurk (Radinsky et al., 2011) and MEN (Bruni et al., 2013) datasets. However, these datasets are limited, since they contain only nouns (except MEN). When using word similarity in pattern ranking schemas, we will likely want to measure similarity between nouns, verbs, adjectives and adverbs. Still, these datasets provide a good starting point for evaluation of word similarity.

5 Word similarity in pattern ranking

The hypothesis of our research is that the use of word similarity will allow better pattern ranking to better prevent semantic drift. We face three main challenges in supporting this hypothesis. First, we need to understand the interdependence of the three elements presented in the three previous sections: pattern representation, pattern confidence estimation, and word similarity. Second, we need to devise an appropriate set-up to perform our bootstrapping approach. Third, we need to properly evaluate the role of the different variations in preventing semantic drift.

An important exploration will be to decide where the word similarity has the largest potential. For example, in the work of Stevenson and Greenwood (2005), similarity is directly applied on parts of the triples found (Subject, Verb predicate or Object), or in the work of Zhang et al. (2014), word similarity would be integrated in the matching and similarity functions over dependency trees, instead of using string equality.

As we see, the integration of word similarity measures would be different depending on the type of pattern representation used. Furthermore, in some representation, there is already a notion of pattern generalisation, such as in the work of Pasca et al. (2006), where words are replaced with more general classes, if any. In such case, word similarity measures are used at the core of the pattern representation, and will further impact pattern ranking.

As we will eventually be building a complex system, we intend to follow a standard methodology of starting with a baseline system for which we have an evaluation, and then further evaluate the different variations to measure their impact. As the number of combination of possible variations will be too large, time will be spent also on partial evaluation, to determine most promising candidates among word similarity measures, and/or pattern representation and/or pattern confidence estimation, to understand strength and weaknesses of each aspect independently of the others.

Our proposed methodology is to take promising ranking approaches among the one presented in Section 3, and promising pattern representations from what was presented in Section 2. We can evaluate their combined performance through N different iteration intervals and incorporate different similarity measures (some best measures chosen from the evaluation on known datasets) to measure the performance of the system.

As our baseline system, we are inspired by CPL subsystem of NELL (Carlson et al., 2010) since it is one of the largest, currently active, bootstrapping system in the literature. As in NELL, we will use ClueWeb[1] as our corpus, and for the set of relations, we will use the same seed instances and relations as in the evaluation of NELL (Carlson et al., 2010).

As for the bootstrapping RE system, to evaluate the precision, we will randomly sample knowledge from the knowledge base and evaluate them by sev-

[1] http://www.lemurproject.org/clueweb09/

eral human judges. The extracted knowledge could be validated using a crowdsourcing application such as MTurk. This method is based on NELL (Carlson et al., 2010). To evaluate its recall, we have to concentrate on already annotated relations. For example, Pasca et al. (2006) evaluates the relation *Person-BornIN-Year*. As a Gold Standard, 6617 instances were automatically extracted from Wikipedia. Instead of measuring recall for specific relation, we could use relative recall (Pantel et al., 2004; Pantel and Pennacchiotti, 2006). We can evaluate our contributions by the relative recall of system A (our system) given system B (baseline).

6 Related issues in pattern ranking

Our main contribution on the impact of word similarity on pattern ranking will necessarily bring forward other interesting questions that we will address within our thesis.

6.1 Choice of seed

As we saw, pattern ranking is often dependent on the comparison of instances found from one iteration to the next. At iteration 0, we start with a seed of instances. We can imagine that the manual selection of these seeds will have an impact on the following decisions. As our similarity measures are used to compare candidate instances to seed instances, and as we will start with NELL seed set, we will want to evaluate its impact on the bootstrapping process.

It was shown that the performance of bootstrapping algorithms highly depend on the seed instance selection (Kozareva and Hovy, 2010). Ehara et al. (2013) proposed an iterative approach where unlabelled instances are chosen to be labelled depending on their similarity with the seed instances and are added in the seed set.

6.2 Automatic selection of patterns

Something noticeable among our surveyed pattern ranking approaches is the inclusion of empirically set thresholds that will definitely have an impact on the semantic drift, but which impact is not discussed. Most authors (e.g (Carlson et al., 2010; Sun and Grishman, 2010; McIntosh and Yencken, 2011; Zhang et al., 2014) among recent ones) select the top-*N* best ranked patterns to be promoted to next iteration. Other authors (Pasca et al., 2006; Dang and Aizawa, 2008; Carlson et al., 2010) select the top-*M* ranked instances to add in the seed set for the next iteration. Other authors (Brin, 1999; Agichtein and Gravano, 2000; Sijia et al., 2013) only apply a filtering step without limiting pattern/instance selection.

In our work, including word similarity within pattern ranking will certainly impact the decision on the number of patterns to be promoted. We hope to contribute in developing a pattern selection mechanism that will be based on the pattern confidence themselves rather than on an empirically set *N* or *M*.

7 Conclusion

In this paper, we have presented our research proposal, aiming at determining the impact of employing word similarity measures within pattern ranking approaches in bootstrapping systems for relation extraction. We presented two aspects of pattern ranking on which the integration of word similarity will be dependent, that of pattern representation and pattern ranking schemas. We showed that there are minimally lexical and syntactic pattern representations on which different methods of generalizations can be applied. We performed a non-exhaustive survey of pattern ranking measures classified in five different categories. We also briefly looked into different word similarity approaches.

This sets the ground for the methodology that we will pursue, that of implementing a baseline bootstrapping system (inspired by NELL, and working with ClueWeb as a corpus), and then measuring the impact of modifying the pattern representation and the pattern ranking approaches, with and without the use of word similarity measures. There is certainly a complex intricate mutual influence of the preceding aspects which we need to look into. Lastly, we briefly discussed two related issues: the choice of seed set and better estimation of number of patterns to promote.

Acknowledgments

This work has seen the day with the help and advice of Caroline Barrière, my research supervisor at CRIM. This research project is partly funded by an NSERC grant RDCPJ417968-11, titled *Toward a second generation of an automatic product coding system*.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the fifth ACM conference on Digital libraries - DL '00*, pages 85–94, New York, New York, USA. ACM Press.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09*, pages 19–27, Morristown, NJ, USA. Association for Computational Linguistics.

Enrique Alfonseca, Pablo Castells, Manabu Okumara, and Maria Ruiz-Casado. 2006a. A Rote Extractor with Edit Distance-based Generalisation and Multi-corpora Precision Calculation. In *COLING-ACL'06 Proceedings of the COLING/ACL Poster Session*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

Enrique Alfonseca, Maria Ruiz-Casado, Manabu Okumura, and Pablo Castells. 2006b. Towards Large-scale Non-taxonomic Relation Extraction : Estimating the Precision of Rote Extractors. In *Proceedings of the second workshop on ontology learning and population, Coling-ACL'2006*, pages 49–56.

Satanjeev Banerjee and Ted Pedersen. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In Alexander Gelbukh, editor, *Computational linguistics and intelligent text processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer Berlin Heidelberg, Berlin, Heidelberg, February.

Sebastian Blohm, Philipp Cimiano, and E Stemle. 2007. Harvesting Relations from the Web - Quantifiying the Impact of Filtering Functions. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1316–1321.

Sergey Brin. 1999. Extracting Patterns and Relations from the World Wide Web. In Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases*, Lecture Notes in Computer Science, chapter Extracting, pages 172–183. Springer Berlin Heidelberg, Berlin, Heidelberg, March.

Elia Bruni, Nam Khan Tran, and Marco Baroni. 2013. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 48:1–47.

Razvan C Bunescu and Raymond J Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Naural Language Processing (HLT/EMNLP)*, pages 724–731, Vancouver, Canada.

Andrew Carlson, Justin Betteridge, and Bryan Kisiel. 2010. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pages 423–429, Morristown, NJ, USA. Association for Computational Linguistics.

VB Dang and Akiko Aizawa. 2008. Multi-class named entity recognition via bootstrapping with dependency tree-based patterns. In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 76–87.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2013. Understanding seed selection in bootstrapping. In *Proceedings of the TextGraphs-8 Workshop*, pages 44–52, Seattle, Washington, USA.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th conference on World Wide Web - WWW '04*, page 100, New York, New York, USA. ACM Press.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 406–414, New York, New York, USA. ACM Press.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pages 415–422, Morristown, NJ, USA. Association for Computational Linguistics.

JJ Jiang and DW Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *In the Proceedings of ROCLING X, Taiwan*, pages 19–33.

Zornitsa Kozareva and Eduard Hovy. 2010. Not All Seeds Are Equal : Measuring the Quality of Text Mining Seeds. In *Proceeding HLT '10 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 618–626.

Claudia Leacock and Martin Chodorow. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database.*, pages 265–283. MIT Press.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries. In *Proceedings of the 5th annual international conference on Systems documentation - SIGDOC '86*, pages 24–26, New York, New York, USA. ACM Press.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304.

T McIntosh and Lars Yencken. 2011. Relation guided bootstrapping of semantic lexicons. In *Proceeding HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, pages 266–270.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pages 113–120, Morristown, NJ, USA. Association for Computational Linguistics.

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *Proceedings of the 20th international conference on Computational Linguistics COLING 04*, pages 771–777.

M Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and A Jain. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. *AAAI*, 6:1400–1405.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web - WWW '11*, page 337, New York, New York, USA. ACM Press.

Philip Resnik. 1995. Using IC to Evaluation the Semantic Similarity in a Taxonomy. In *Proceeding IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 448–453.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October.

Chen Sijia, Li Yan, and Chen Guang. 2013. Reducing semantic drift in bootstrapping for entity relation extraction. In *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, pages 1947–1950. Ieee, December.

Mark Stevenson and Mark A Greenwood. 2005. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, pages 379–386, Morristown, NJ, USA. Association for Computational Linguistics.

Ang Sun and Ralph Grishman. 2010. Semi-supervised Semantic Pattern Discovery with Guidance from Unsupervised Pattern Clusters. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1194–1202, Beijing.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics -*, pages 133–138, Morristown, NJ, USA. Association for Computational Linguistics.

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, volume 1, pages 343–350, Morristown, NJ, USA. Association for Computational Linguistics.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering Relations Between Named Entities form a Large Raw Corpus Using Tree Similarity-based Clustering. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *Natural Language Processing – IJCNLP 2005*, Lecture Notes in Computer Science, pages 378–389. Springer Berlin Heidelberg, Berlin, Heidelberg.

Chunyun Zhang, Weiran Xu, Sheng Gao, and Jun Guo. 2014. A bottom-up kernel of pattern learning for relation extraction. In *The 9th International Symposium on Chinese Spoken Language Processing*, pages 609–613. IEEE, September.