Data-driven Relation Discovery from Unstructured Texts

Marilena Ditta, Fabrizio Milazzo, Valentina Raví, Giovanni Pilato and Agnese Augello
ICAR CNR, Viale delle Scienze Edificio 11, Palermo, 90128, Italy

Keywords: Latent Semantic Analysis, Triplet Extraction.

Abstract: This work proposes a data driven methodology for the extraction of *subject-verb-object* triplets from a text corpus. Previous works on the field solved the problem by means of complex learning algorithms requiring hand-crafted examples; our proposal completely avoids learning triplets from a dataset and is built on top of a well-known baseline algorithm designed by Delia Rusu et al.. The baseline algorithm uses only syntactic information for generating triplets and is characterized by a very low precision i.e., very few triplets are meaningful. Our idea is to integrate the semantics of the words with the aim of filtering out the wrong triplets, thus increasing the overall precision of the system. The algorithm has been tested over the Reuters Corpus and has it as shown good performance with respect to the baseline algorithm for triplet extraction.

1 INTRODUCTION

Nowadays the knowledge contained in the Web has exceeded the human capacity to access it and the use of technology capable of automatically understanding written texts has become mandatory. The major challenge for researchers in automatic text understanding is thus to create tools that can scale to the Web: the basic idea is to transform text in structured information through using semantic contents.

Traditionally, Information Extraction (IE) systems focus on the identification of pre-specified target relations between entities, such as person, organization or location, or between tuples/pairs of common nominals (Bach and Badaskar, 2007). Such approaches use language processing algorithms either supervised or semi-supervised tuned for the domain of interest and they are able to extract tuples of words linked by certain kind of relationships. In particular, the task of *semantic relation extraction* consists in recognizing tuples in the form $(e_1, e_2, \ldots, e_n)$ where e_i are entities or common nominals in a predefined relation r. To this aim, Named-Entity Recognizers (NERs) (Nadeau and Sekine, 2007; Richard Socher and Ng., 2015a) or dependency parsers (Kubler et al., 2009; Richard Socher and Ng., 2015b) may be used to ease relation extraction: as an example, in the sentence "*The bowl contained apples, pears and oranges*" the nominals (*bowl, pears*) and (*bowl, apples*) are connected in a *contained-in* relation.

The past literature proposed to solve the relation extraction task as a binary classification problem (Bach and Badaskar, 2007): a labeled training set of sentences containing entities in a given relation r is provided as input to the system; any new sentence s is thus processed by the classifier which provides a binary output specifying whether the entities in s are linked by r or not. The main drawback of these approaches consists in the severe human intervention required to create new hand-crafted extraction rules or new hand-tagged training examples; even worse, these approaches do not scale to wide heterogeneous corpora such as the Web, where target relations are not known in advance.

The Open Information Extraction (OIE) paradigm, has been developed in order to overcome the limits imposed by supervised and semi-supervised approaches; OIE avoids relation specificity (Shinyama and Sekine, 2006), does not need domain-specific training data and scales to wide heterogeneous corpora such as the Web (Etzioni et al., 2008). An OIE system, in a preliminary stage, learns , for a language, a syntactic model of the sentences where two entities are linked by a relation;in a subsequent stage, the corpus is given as input to the system, which returns a set of tuples (i.e., entities or nouns and adjectives linked by a relation) on the basis of the learned model. Such tuples are usually generated in a *verb-based* form: as an example in the sentence "*Several investors form an alliance in a hostile takeover.*", the system extracts the tuple (*investors, form, alliance*). The works (Soderland et al., 2010) implemented a prototype of an OIE system, by using a noun-phrase chunker and a

Ditta, M., Milazzo, F., Ravì, V., Pilato, G. and Augello, A..
Data-driven Relation Discovery from Unstructured Texts.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 597-602
ISBN: 978-989-758-158-8

relation generator to extract the possible relations for each sentence; in a second step, the system maps each relation into a feature vector and trains a Naïve Bayes model. The Reverb system (Fader et al., 2011) is an evolution of such former works and is implemented as an extractor for verb-based relations which uses a logistic regression classifier trained with syntactic features. The extracted relations are filtered by two kinds of analysis: *syntactic* and *lexical*; the syntactic analysis requires the constituents of the relation to match a pre-defined set of POS tags patterns. The lexical analysis filters out overly-specific relations by looking to the frequency of their constituents over the considered input corpus.

Finally, the work (Rusu et al., 2007), (Atapattu Mudiyanselage et al., 2014) and (Ceran et al., 2012) propose a rule-based approach for the extraction of subject-verb-object triplets from unstructured texts: the parsing tree of the input sentence is browsed and the subject is computed as the first noun found in the noun-phrase, the verb is the deepest leaf found in the verb-phrase, while the object is the first noun/adjective found in a phrase sibling of the verb-phrase. Such an approach is very fast if compared to its predecessors as it does not require learning pre-defined relations; on the other hand its precision seems to be quite low (as shown further in the experimental section) and this is due to the use of only the POS Tagging information.

The main contribution of this paper is the development of a methodology to extract meaningful triplets from unstructured texts. We chose to build our methodology on top of the baseline algorithm developed by Rusu et al. (Rusu et al., 2007): such as algorithm, if compared to other state-of-the-art methods, needs very low computational requirements, avoids learning and does not need hand-tagged examples. Such baseline algorithm, on the other hand, uses only syntactic information to extract relations and this may lead to a low quality of the generated triplets (poor precision/recall); to this aim we propose to integrate *Latent Semantic Analysis* (Landauer and Dutnais, 1997; Deerwester et al., 1990), whose statistical foundation has been recently explained in (Pilato and Vassallo, 2015) in order to filter out low quality triplets thus improving precision/recall. At a preliminary stage, pairs of tightly semantically related words are extracted from the corpus; in a further stage the sentences containing such words are parsed and syntactically analyzed to discover the linking relations. As this is a preliminary work, we will focus only on the extraction of first-order relationships i.e., *triplets* in the form (*subject, verb, object*).

2 THE PROPOSED APPROACH

The main purpose of this work is to extract a set of triplets of words in a *subject-verb-object* form from a documents corpus; the proposed algorithm does not require any pre-defined relation and tries to discover valid relations through the analysis of the semantic of words. Firstly, the algorithm builds a semantic space by means of the Latent Semantic Analysis (*LSA*) in order to reveal *pairs* of words which are somehow related each other. Any pair of words in a tight semantic association (high values of the cosine similarity in the semantic space) is then expanded in a triplet form (if possible) by looking into the corpus for a verb binding that pair: i.e., the algorithm looks for $s-v-o$ triplets.

The triplets extraction task is accomplished by four main blocks: *a) Micro-documents Extraction:* extracts sentences from the corpus and records them as micro-documents; *b)Word-Tags-Documents Generation:* pre-processes the micro-documents and creates a list of unique words; each word will be associated to a *tagset* representing its different part of speech (*POS*) tags as well as to its frequency counts in the extracted micro-documents corpus; *c) Pairs Extraction:* builds the LSA semantic space and selects relevant pairs of words semantically related; *d) Triplet Generation:* tries to match the relevant pairs into $s-v-o$ triplets extracted from the micro-documents.

2.1 Micro-documents Extraction

The aim of this block is to segment the input corpus in a sequence of sentences. As described in Figure 1, the *sentences-extractor* module reads the corpus and produces a sequence of sentences (i.e., a sequence of words included between two periods). *Case-folding* is applied to the sentences. Each sentence is therefore saved in a text file.

2.2 Word-Tags-Documents Generation

As shown in Figure 2, the extracted micro-documents are tokenized, tagged and lemmatized in order to reduce the dimensionality of the term-document matrix that will be provided as input of LSA; stop-words are also removed. After such pre-processing steps, the *dictionary extractor* module associates to each distinct word $w_i \in W$ its related tagset $tags_i \in T$ and the frequency counts $docfreq_{i,j}$ of the word w_i in each micro-document $d_j \in D$, where W,T,D are respectively the sets of unique words, tags and micro-documents. The output of the block is a dictionary data structure $WTD = \{w_i, tags_i, docfreq_{i,j}\}$.

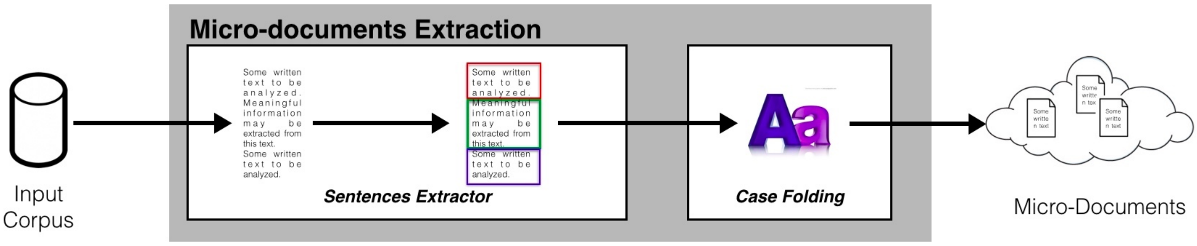

Figure 1: Micro-Documents Extraction block. The input corpus is split into sentences saved as text-files.

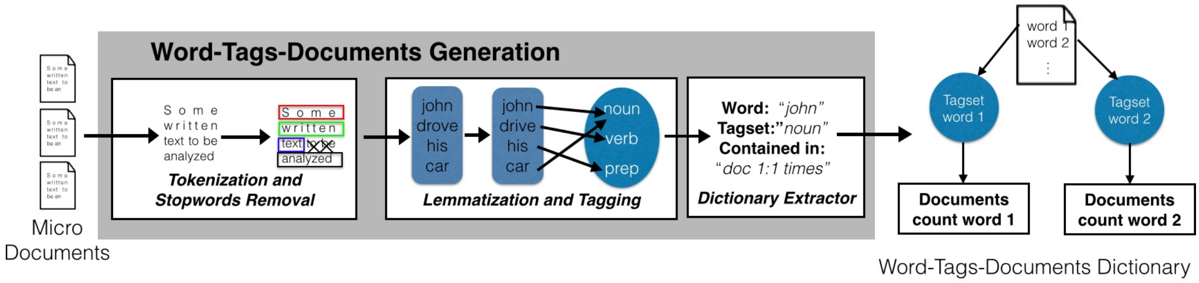

Figure 2: Word-Tags-Documents Generation block. Each word is associated to a tag-set and to a frequency vector.

2.3 Pairs Extraction

The Pairs Extraction block, depicted in Figure 3, accepts as input the WTD structure and provides as output a list of pairs of semantically related words of the type *(noun, noun)* or *(noun, adjective)*.

The words belonging to the WTD structure are initially filtered with respect to their frequency distribution, in order to reduce noise. In particular, a word-frequencies histogram is built by computing the value $c_i = \sum_j^J docfreq_{i,j}$ for each word w_i; the words are thus sorted according to their c_i value. Finally, only a percentage ϕ of the distribution is retained so that less frequent words are filtered-out and removed from the WTD dictionary.

The next step involves computation of LSA which projects the filtered words onto a dimensionally-reduced semantic space (Deerwester et al., 1990) where semantically similar terms lie near each other.

The word-document matrix $\mathbf{A} \in R^{|W|,|D|}$ is built by linking W to D and then it is factorized in three matrices: $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\mathbf{T}$ where:

- $\mathbf{U}$ is a $|W| \times r$ matrix;
- $\mathbf{V}$ is a $|D| \times r$ matrix;
- Σ is a $r \times r$ diagonal matrix, containing the square roots of the eigenvalues of $\mathbf{AA^T}$.

Truncated SVD considers only the largest $k << r$ singular values of Σ, and removes the least significant dimensions: $\tilde{\mathbf{A}} = \mathbf{U_k}\Sigma_k\mathbf{V_k}^\mathbf{T}$. Words are represented as points in such $k-$dimensional space which is used to evaluate semantic affinity i.e., the degree of connection in terms of meaning, with their topological proximity in the semantic space. The output terms are filtered by the tag-filtering module, which produces a set of candidate words CW i.e., all those words w_i labelled as noun or adjective by the tagging module. Thereafter, the cosine distance $cd_{i,j}$ between all the words $(w_i, w_j) \in CW \times CW$ is computed; the relevant pairs RP will be selected so that $cd_{i,j} > \theta$, where θ is a proper threshold.

2.4 Triplets Generation

The Triplets Generation block, shown in Figure 4 accepts as input the RP set, the WTD structure and the micro-documents and produces as output a set of triplets in the form $s - v - o$.

For each pair of words belonging to RP only the micro-documents containing both the words are selected by the *document intersection* block. In order to extract a triplet of the form *subject-predicate-object*, such micro-documents are syntactically parsed in order to obtain a *treebank* representation (Surdeanu, 2015). The *Triplet extraction* module is implemented by the algorithm presented in (Rusu et al., 2007), which will be named as *Blind Triplet Extraction* (BTE) in the following. Such algorithm runs along the tree of each sentence and identifies the subject as the first noun found in the NP subtree; a further search detects the deepest verb in VP subtree; finally, the object is extracted as the first noun/adjective lying in the siblings of the subtree containing the verb. The *triplet matching* module tries to match the *s-o* portion of the

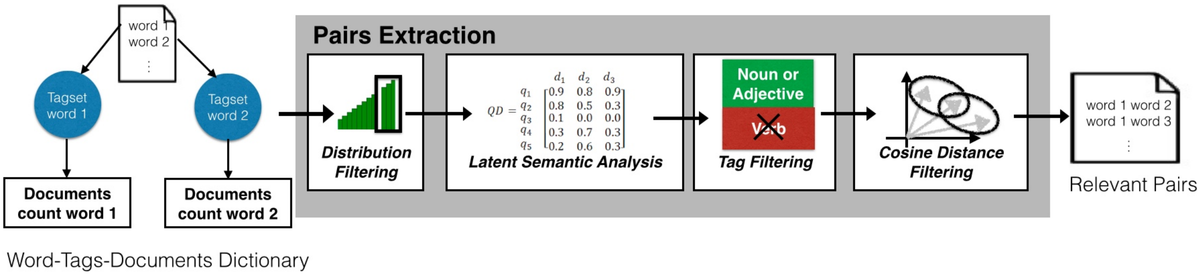

Figure 3: Pairs Extraction Block. A set of pairs subject-object is provided as output.

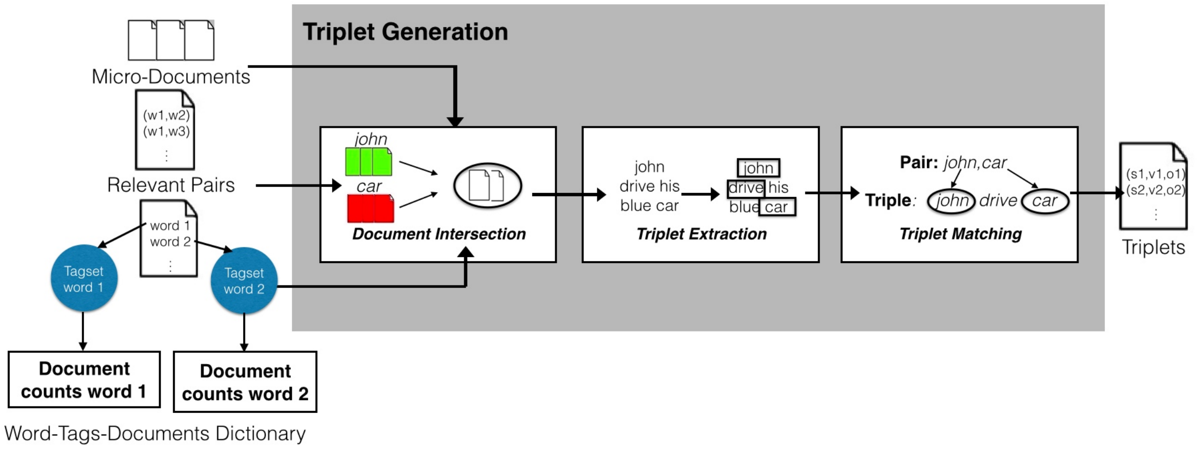

Figure 4: Triplet Generation block. Relevant pairs are matched with the (s-v-o) triplets found in micro-documents.

generated triplets with the pairs contained in *RP*; if a match is found, that triplet is recorded and provided as output of the block.

3 EXPERIMENTAL EVALUATION

The aim of this Section is to evaluate the performance of the proposed algorithm under several different parameters settings. We extracted 150 KBytes of text from Reuters Corpus (Rose et al., 2002) and generated 1640 micro-documents (sentences) which give rise to a total amount of 2864 distinct words; after running the blind triplet extraction (BTE) algorithm described in (Rusu et al., 2007), 941 triplets in the form *subject-verb-object* were extracted. A team of three experts validated the 941 triplets and classified 436 as syntactically correct and 505 as wrong.

Different external tools were used to implement the modules depicted in Figures 1,2,3 and 4:

- the *lemmatization* and *tagging* modules were implemented by using TreeTagger (Schmid, 1995);
- The LSA module was implemented by using the *SVDLIBC* package (Berry et al., 2015);
- The triplet extraction module was implemented by using Stanford-Core-NLP (Manning et al., 2014), which allowed for building the parsing trees, while the BTE algorithm was used to extract the *s-v-o* triplets.

In our experiments we set the number of LSA components to $k \in \{70, 150\}$; the tag-filtering module was set to recognize all the words tagged as noun: $\{NN, NNS, NP, NPS\}$ or as adjective: $\{JJ, JJR, JJS\}$ so the candidate subject-object pairs, provided by the pairs extraction block, were all of the types: *noun-noun*, *noun-adjective* and *adjective-adjective*. The threshold of the distribution filtering module was set to $\phi \in \{0.5, 0.75, 0.85, 1.0\}$ and finally, the cosine distance filtering threshold θ ranged in the interval $[-1.0, 1.0]$ with a step equal to 0.1.

Figures 5 and 6 provide a brief comparison of the performance achieved by our algorithm against BTE. As expected, BTE achieves the worst performance in terms of precision ($\sim 46\%$) which is also independent on the recall value as the extraction method does not take into account the statistical properties of the triplets constituents. As it can be easily seen from the figures, the more the infrequent words are discarded (i.e., ϕ decreases), the more the average precision increases. Also, while the θ threshold increases then recall decreases (less triplets are selected by our al-

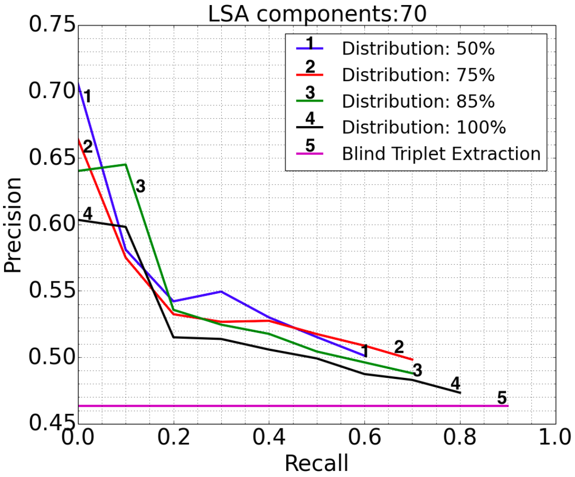

Figure 5: Average Precision as a function of the recall, for k=70 components of the LSA.

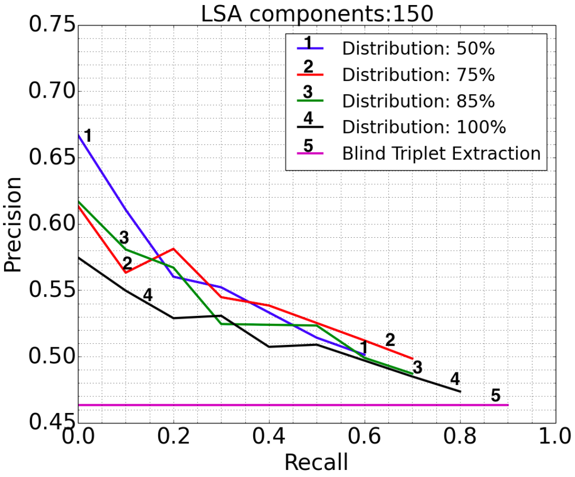

Figure 6: Average Precision as a function of the recall, for k=150 components of the LSA.

gorithm) but, at the same time, the related precision increases (high values of the cosine distance represent tightly related constituents). In the best case, the average precision approaches 70% in the case of $\phi = 0.5, k = 70$ which drops to 50% in the worst case where $\phi = 1.0, k = 150$; as a final remark, the performance, for this experiments set, appears to be slightly better for $k = 70$ components of the LSA and this is probably due to the small number of the involved micro-documents.

4 CONCLUSIONS

This work presented an unsupervised data-driven methodology to relation extraction from text corpus. Unlike other works, our methodology does not require learning pre-defined relations and discovers relations by exploiting a mixed syntactical-semantic approach. The baseline algorithm needed very low computational requirements but was characterized by very low precision. In contrast, experimental results demonstrated that our methodology shows a good degree of precision if compared to the baseline triplet extraction algorithm although the respective recall stayed quite low. As a future work we are planning to discover high-order relationships where more than two constituents are involved in the relations; moreover the use of our methodology may be investigated as a tool to support the automatic creation of ontologies by generating sets of RDF triplets. Finally, we are currently investigating the performance of our proposal over the ClueWeb09 Dataset (Callan, 2009) which is made up of about 1 billion pages and 604 million triplets.

REFERENCES

Atapattu Mudiyanselage, T., Falkner, K., and Falkner, N. (2014). Acquisition of triples of knowledge from lecture notes: a natural language processing approach. In *7th International Conference on Educational Data Mining (04 Jul 2014-07 Jul 2014: London, United Kingdom)*.

Bach, N. and Badaskar, S. (2007). A Survey on Relation Extraction.

Berry, M., Do, T., O'Brien, G., Krishna, V., and Varadhan, S. (2015). Svdlibc. http://tedlab.mit.edu/∼dr/svdlibc/.

Callan, J. (2009). The clueweb09 dataset. http://www.lemurproject.org/clueweb09.php/.

Ceran, B., Karad, R., Mandvekar, A., Corman, S. R., and Davulcu, H. (2012). A semantic triplet based story classifier. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 573–580. IEEE.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kubler, S., McDonald, R., Nivre, J., and Hirst, G. (2009). *Dependency Parsing*. Morgan and Claypool Publishers.

Landauer, T. K. and Dutnais, S. T. (1997). A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30(1):1–20.

Pilato, G. and Vassallo, G. (2015). Tsvd as a statistical estimator in the latent semantic analysis paradigm. *Emerging Topics in Computing, IEEE Transactions on*, 3(2):185–192.

Richard Socher, John Bauer, C. D. M. and Ng., A. Y. (2015a). Stanford ner. http://nlp.stanford.edu/software/CRF-NER.shtml.

Richard Socher, John Bauer, C. D. M. and Ng., A. Y. (2015b). Stanford parser. http://nlp.stanford.edu/software/lex-parser.shtml.

Rose, T., Stevenson, M., and Whitehead, M. (2002). The reuters corpus volume 1 - from yesterdays news to tomorrows language resources. In *In Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 29–31.

Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Triplet Extraction From Sentences. *Proceedings of the 10th International Multiconference "Information Society - IS 2007*, A:218–222.

Schmid, H. (1995). Treetagger a language independent part-of-speech tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. pages 304–311.

Soderland, S., Roof, B., Qin, B., Xu, S., Etzioni, O., et al. (2010). Adapting open information extraction to domain-specific relations. *AI magazine*, 31(3):93–102.

Surdeanu, M. (2015). Penn treebank project. http://www.surdeanu.info/mihai/teaching/ista555-spring15/readings/PennTreebankConstituents.html.